
Sous la direction de :
Camille Paloque-Berges, Christophe Masutti

Histoires et cultures du Libre

Des logiciels partagés aux licences échangées



Framasoft a été créé en novembre 2001 par Alexis Kauffmann. En janvier 2004 une association éponyme a vu le jour pour soutenir le développement du réseau. Pour plus d'information sur Framasoft, consulter <http://www.framasoft.org>.

Se démarquant de l'édition classique, les Framabooks sont dits « livres libres » parce qu'ils sont placés sous une licence qui permet au lecteur de disposer des mêmes libertés qu'un utilisateur de logiciels libres. Les Framabooks s'inscrivent dans cette culture des biens communs qui, à l'instar de Wikipédia, favorise la création, le partage, la diffusion et l'appropriation collective de la connaissance.

Le projet Framabook est coordonné par Christophe Masutti. Pour plus d'information, consultez <http://framabook.org>.

Copyright 2013 : Camille Paloque-Berges, Christophe Masutti, Framasoft (coll. Framabook)

Histoires et cultures du Libre. Des logiciels partagés aux licences échangées est placé sous licence Creative Commons -By (3.0).

Édité avec le concours de l'INRIA et *Inno*³.

ISBN : 978-2-9539187-9-3

Prix : 25 euros

Dépôt légal : mai 2013, Framasoft (impr. *lulu.com*, Raleigh, USA)

Pingouins : LL de Mars, Licence Art Libre

Couverture : création par Nadège Dauvergne, Licence CC-By

Mise en page avec L^AT_EX



Cette œuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution 2.0 France.

<http://creativecommons.org/licenses/by/2.0/fr>



Préface

Hervé LE CROSNIER

Aborder un sujet par la perspective historique est toujours une source de clarification. L'histoire offre un éclairage qui prend en compte la dimension temporelle et l'environnement social, économique et culturel. Ce livre en est une preuve supplémentaire, et il faut remercier Camille Paloque-Berges et Christophe Masutti d'en avoir pris l'initiative et d'avoir réuni ici toutes ces contributions. L'informatique est récente, une soixantaine d'années. Les changements et l'impact sur le monde sont pour autant majeurs. Du calculateur électronique à la machine à communiquer mondiale d'aujourd'hui, c'est non seulement un chemin technique, mais également l'irruption d'une nouvelle structure sociale mondiale qui est en jeu. Les informaticiens ont changé de statut dans le cours de ces années. De serviteurs de la machine, ils deviennent les réalisateurs d'idées et de concepts destinés à changer le monde. Les utopies numériques qui ont fleuri tout au long de ce parcours se transforment en pratiques sociales. Plus de deux milliards d'humains communiquent en instantané dans toutes les langues. La moitié d'entre eux gèrent leur sociabilité sur un même réseau privé. Ces deux facettes portent en elles à la fois la magie qu'a introduit le numérique dans la vie quotidienne, ce sentiment d'appartenir à un monde commun ; et les dangers de nouvelles monopolisations, de pouvoirs disproportionnés et de traçage des activités privées.

C'est dans ce monde ambivalent, qu'ils ont contribué à construire, que les informaticiens ont du faire des choix au-delà de leur technique de spécialité. Des choix philosophiques, guidés par leur conception du monde, mais également par le plaisir collectif qu'ils peuvent prendre à partager leur code, leur savoir-faire, leurs intuitions techniques. Affirmons-le d'emblée, les informaticiens prennent du plaisir à coder. Pour beaucoup, et peut-être pour les historiens des techniques qui liront ce livre, cela relève d'un autre monde. Les caricatures médiatiques du *no-life*, isolé derrière son écran, ne collent pas avec le type de relations qui ont pu se tisser dans les groupes de *geeks*. Car le plaisir de la programmation et des usages pionniers de l'Internet ont incité à la création d'un groupe culturel qui se forge ses propres référents, fait émerger ses héros, installe ses sociabilités, ses marques de respect. La culture hacker a favorisé la diffusion des idées proprement révolutionnaires du logiciel libre. Car, comme toute culture, elle se base avant tout sur l'échange, sur la capacité à partager ses centres d'intérêt, à créer des cercles informels. À commencer par le partage du code informatique.

Le logiciel est né libre. Dans les années cinquante et soixante, le marché de l'informatique ne concernait que le matériel et le service. Le logiciel était simplement le moyen de faire fonctionner ou de donner une utilité à ces machines qui pouvaient dévorer les chiffres et rendre calculatoire toutes les autres activités. Au point, comme nous le rappelle l'histoire des jeux vidéos présentée ici par Damien Djaouti, que les jeux eux-mêmes devenaient pour les constructeurs de matériel des *proofs of concept*, capables de valider le fonctionnement d'un matériel informatique. La programmation permet d'inscrire des idées et des concepts dans la mémoire culturelle des ordinateurs. Au-delà de la conception d'un programme permettant une activité sur un ordinateur donné, le logiciel est un moyen de traiter, sur cet ordinateur particulier, un projet plus général. Le programmeur espère que ses modèles, une fois mis en œuvre sur une machine, seront répliqués sur d'autres ordinateurs, et deviendront des pensées communes. La comparaison, menée ici par Pascal Robert, entre les idées de J.C.R. Licklider sur la communication et celles de Douglas Englebart sur la convivialité des interfaces est éclairante. Dans le legs de ces deux figures majeures de l'informatique des années soixante, nous trouverons la souris et l'Internet.

C'est-à-dire les germes du modèle de l'informatique qui est aujourd'hui pleinement partagé et constitue la base culturelle commune.

Mais en chemin, les idées que ces ingénieurs et chercheurs diffusaient tous azimuts, sont revenues bardées de brevets, de code propriétaire, accompagnées d'une palanquée d'avocats, au point qu'actuellement, Apple et Google dépensent plus en frais juridiques qu'en recherche et développement. Le modèle universitaire, qui consiste à rendre disponible le travail pour tout usage, y compris privé, et même éventuellement privé, était celui des chercheurs pionniers. Il a joué un rôle majeur dans l'extension de l'informatique. Dès 1984, le fait que l'UNIX de Berkeley comprenait en standard une pile TCP/IP a permis aux laboratoires universitaires de tester l'Internet, d'intégrer le réseau dans la recherche. Le fait que les normes de l'Internet aient été élaborées collectivement et publiquement, a permis la diffusion très rapide des concepts et des modèles parmi les ingénieurs. Cependant, le chemin vers l'appropriation privée des travaux universitaires, à l'image de Mac OS X utilisant l'UNIX de Berkeley, mais privatisant les propres apports de Apple, a rompu la chaîne de transmission des savoirs informatiques.

Or, dans l'éthique des hackers, la transmission, la capacité à voir et comprendre les subtilités d'un programme, à décrypter le fonctionnement d'un ordinateur, sont des valeurs principales, comme le rappelle Steven Levy dans son ouvrage qui vient d'être traduit en français [*L'éthique des hackers*, Globe, 2013], et qui est le plus cité par les auteurs ici. Le hacker a besoin de comprendre, d'apprendre pour améliorer. Cela renvoie à la nature proprement incrémentale du logiciel. Une fois l'architecture mise en place, un logiciel doit évoluer en permanence. D'abord pour s'adapter aux changements de l'environnement informatique, aux nouvelles machines, comme aux nouveaux concepts de programmation ou d'interface homme-machine. Ensuite pour répondre aux attentes des utilisateurs. Plusieurs articles de ce livre insistent sur la place des usagers dans les communautés ou les écosystèmes autour d'un logiciel ou d'un service. L'informaticien travaille sur un modèle du monde, qu'il va faire manipuler par la machine, et sur un modèle de l'utilisateur qui définit l'ergonomie générale et les options possibles. Les modèles ne correspondent jamais au monde réel. Il s'agit toujours de versions simplifiées. Or, l'informatique a ce charme fabuleux d'être une technique plastique, modulaire et modulable, réversible,

adaptable. Le grand jeu n'est jamais fini. Plus les usagers feront remonter d'informations sur leurs besoins, sur les problèmes rencontrés, plus les usages eux-mêmes dévoileront des failles, des erreurs, des bugs, et plus les capacités des groupes de programmeurs à faire évoluer le logiciel seront grandes. C'est également ce qui explique que, parmi les millions de logiciels libres disponibles, certains émergent et perdurent grâce à cette symbiose nouvelle entre les développeurs et les utilisateurs. C'est notamment le cas des logiciels libres d'infrastructure, depuis le gestionnaire de nom de domaine BIND qui est au cœur de l'Internet, jusqu'aux serveurs web, notamment Apache, les bases de données PostgreSQL et MySQL (et maintenant les bases noSQL), et bien évidemment les systèmes d'exploitation, comme GNU/Linux et ses différentes distributions. Les usagers qui vont constituer les communautés d'appui ne sont pas directement le grand public, mais les autres informaticiens qui utilisent ces outils pour produire des services au public. La garantie que cette chaîne complète – cet écosystème – va perdurer est au cœur de la question des licences qui, depuis très longtemps, crée un fort débat parmi les tenants du Libre.

Le vaste panorama des licences proposé ici par Benjamin Jean nous aide à y voir plus clair. La force du livre que vous avez dans les mains est de ne jamais sacrifier à une vision partisane. Comme dans toute activité qui appelle un investissement humain fort, des conceptions du monde et donc des projets philosophiques, les débats au sein du monde du Libre sont souvent rageurs, toujours engageants, et les excommunications fréquentes. Cela fait partie de la capacité de ce mouvement à structurer une pensée, des groupes, et donc à agir pour étendre cette liberté informatique, si nécessaire au moment même où le numérique et les réseaux prennent une part majeure dans la ré-organisation du monde. Cette vivacité des débats fait vivre l'utopie d'un monde où le savoir, et notamment le savoir sur l'information, ses structures et les outils de sa manipulation, seraient perpétuellement ouverts et libres. Mais la clarté et le recul de l'analyse sont tout autant essentiels. La force, encore une fois, de l'approche historique est de permettre de comprendre la relation entre les points de vue et les moments et situations qui les ont vu naître. Les points de vue ont plus d'importance quand ils ne sont pas encore cristallisés en positions et en guerre de tranchées. Quand tout est encore possible. Reprendre le fil des licences, mesurer leur diversité, et pour chacune les raisons de cette diver-

sité, les tactiques et les orientations que l'on peut en déduire, est pour l'activiste du Libre d'aujourd'hui un document précieux. Nous n'en sommes pas arrivé là par hasard... et il est vraisemblable que la multiplication des licences n'est pas finie. Notamment parce que l'esprit du Libre s'étend toujours au-delà de la sphère informatique et, ce faisant, rencontre des domaines, des types d'information, des communautés et des pratiques qui sont différents. Les licences en sont dès lors différentes, adaptées, comme nous le rappellent Primavera de Filippi et Isabelle Ramade en présentant l'héritage du logiciel libre dans les licences des Creative Commons. Les données, la cartographie, l'accès libre aux publications scientifiques, les encyclopédies coopératives, la production matérielle dans les FabLab... toutes ces activités qui sont la descendance assumée et revendiquée du mouvement des logiciels libres ont besoin de licences qui leurs soient spécifiques, adaptées aux buts, aux objets et aux communautés.

Le coup d'envoi de cette course aux licences aura été la décision fondatrice de Richard Stallman de graver dans le marbre d'un texte juridique les définitions de la liberté informatique. C'est un *hack* majeur : pour construire une alternative dans le cadre actuel, il faut s'appuyer sur le droit et le détourner pour qu'il réponde aux objectifs philosophiques du Libre. Ce *hack* juridique initié par Stallman et Moglen avec la création de la GPL est une leçon révolutionnaire pour de nombreux secteurs de la société. Au-delà même de la sphère du numérique, comme le montrent les mouvements paysans confrontés à la mainmise des semenciers et de leur « propriété intellectuelle » et engageant une activité autonome de gestion et d'échange des semences fermières. Cet héritage n'est pas anodin : dans le tour de passe-passe juridique de la GPL, on trouve la volonté d'inscrire l'indépendance, la protection du bien commun créé collectivement. Cette logique virale, ou *copyleft*, privilégie la protection de la communauté et l'engagement des individus qui y participent plus encore que le produit logiciel lui-même. Ce qui, au fond, devrait être le cas pour toutes les lois qui devraient mettre la liberté des humains devant les règles du commerce.

Mais la liberté n'est pas toujours la tasse de thé des grandes entreprises, qui ont fondé leur pouvoir sur le secret, la propriété et la concurrence. Comment, dans un secteur marqué par de nombreuses entreprises, par une dynamique économique très intense, concilier la liberté et l'entrepreneuriat ? Comment attirer les entreprises dans cette logique d'ouverture de

l'informatique, de transmission et de partage des connaissances ? Libre ou ouvert, voici un débat qui anime, depuis au moins quinze ans et la parution du pamphlet « La Cathédrale et le Bazar » de Eric Raymond, le Landerneau des activistes du logiciel. Un débat est ici remis en perspective : quelles sont les descendances, les succès et les réalisations concrètes des deux approches ? Il ne s'agit jamais de distribuer des bons ou mauvais points, mais de donner au lecteur les outils lui permettant de décrypter les positions, les projets et les actions de ces développeurs qui ont choisi la voie d'une ouverture du code et du partage des savoirs informatiques. Ces deux voies sont choisies par ceux qui s'opposent à la mainmise des logiciels privateurs, des formats exclusifs, des matériels incompatibles. Quand elles sont examinées au prisme de leur histoire, elles n'apparaissent plus si opposées que l'activisme veut le dire, mais au contraire perpétuellement se fécondant l'une l'autre.

Libre, le logiciel est bien évidemment accessible, donc son code source lisible ; il est également modifiable, adaptable ; et son partage est non seulement autorisé, mais de surcroît une dimension absolument nécessaire pour créer un écosystème capable de soutenir son développement. Mais doit-il être viral ou reposer sur un modèle de productivité obtenu par l'ouverture du code ? En bref, s'agit-il de produire du code correct, ré-utilisable, laissant toutes les libertés aux ré-utilisateurs ? Ou bien s'agit-il de construire un univers, un monde, un écosystème qui deviendrait une zone libre, détachée des pouvoirs, et garantissant que jamais il ne sera récupéré et recyclé dans la grande machine marchande qui jusqu'à présent a noyé toutes les velléités de libération ? De créer un commun de la connaissance ? On le voit, le débat porte sur les conceptions culturelles, les référents philosophiques et les projets d'avenir. Et les références à Richard Stallman qui traversent évidemment tout ce livre montrent également le caractère pragmatique de sa philosophie.

L'approche historique, en replaçant les débats dans leur contexte général, permet de revenir aux valeurs fondamentales qui animent les activistes et les acteurs de cette grande scène de l'émergence d'une informatique libre. Et dans un monde qui apparaît souvent clairement pragmatique, fait de bits et de langages d'instruction, les questions philosophiques sur la valeur et l'étendue de la liberté sont un véritable rayon de soleil. Car, et c'est une autre force du livre que vous avez dans les mains, celui-ci traite

de tous ces domaines pour lesquels le logiciel libre a ouvert des logiques de liberté, largement au-delà de l'informatique. L'exemple de Wikipédia, et les débats autour des licences en fonction de l'histoire même de cette encyclopédie coopérative, que présente ici Adrienne Alix, est pleinement significatif.

L'histoire sert bien évidemment à parler du présent pour influencer l'avenir. Et ce livre n'y échappe pas. On ressent le besoin de défendre la neutralité de l'Internet après le survol de l'histoire de ce réseau par Valérie Schafer, débat actuel s'il en est. On comprend mieux les stratégies économiques nouvelles que le Libre fait naître, et les impacts qu'ils peuvent avoir sur d'autres modes de production coopérative après les mises en perspective de toute la partie économique de ce livre. Quand la vente de logiciels fermés ne peut conduire qu'à l'escalade des versions et la multiplication de formats non-normalisés, le Libre vise à défendre des écosystèmes, au sein desquels des communautés, ou des entreprises, peuvent ajouter services, *add-ons* et usages, et renforcer le respect des normes d'interopérabilité. Il y a bien un monde économique dans la construction de communs par le logiciel libre. L'informatique est une plateforme sur laquelle s'échangent d'autres activités, depuis la culture jusqu'au commerce et aux données publiques, et à ce titre il est nécessaire de garantir l'ouverture des logiciels, des formats, des interfaces d'application.

Enfin, l'histoire a pour but de collationner des documents, des témoignages, de donner aux penseurs, aux philosophes et aux activistes des références sur le déroulement et les enchaînements pour leur permettre d'imaginer les utopies et les actions qui vont porter le monde un peu plus loin. De ce point de vue, ce livre est une mine pour les penseurs à venir. Les études de cas, les renvois aux pratiques et aux projets, tels qu'ils ont été portés ou subis, forment un tissu dense. Chaque sujet traité, depuis l'histoire du navigateur Firefox ou du logiciel musical PureData, jusqu'à l'introduction du libre au CNRS et le projet PLUME, rappelle des débats plus généraux. Les exemples servent à renforcer la description de la toile de fond, et pointent des périodes d'inflexion dans l'histoire de l'informatique et des réseaux.

À ce titre, ce livre doit représenter pour les historiens une incitation à créer un secteur de recherche, des réseaux de chercheurs, des publications et des projets autour de l'histoire de l'informatique et de son impact

social. Il commence à exister un fonds documentaire, des témoignages, des expériences qui permettent de suivre à la trace les idées, les concepts et leur évolution comme leur dispersion dans l'ensemble de la société. Il serait trop dangereux de laisser la parole sur l'informatique aux seuls adeptes du *storytelling* pour renforcer leurs places et leurs pouvoirs. Les discours de la techno-béatitude, portés par les services marketing et relayés par les décideurs ont besoin de se confronter à la réalité des succès et des échecs de l'informatique et des secteurs qui s'appuient sur l'informatique et les réseaux. Il s'agit de collationner les sources et de les lire avec suffisamment de recul pour les faire parler, et assez de vision et de mise en perspective pour qu'elles nous apprennent ce que les témoignages isolés et les prises de position particulières ne peuvent donner : une vision synthétique des changements culturels, philosophiques et sociaux portés par l'informatique dans le dernier demi-siècle.

Ce livre est un exemple vivant de ce que cette approche historique peut apporter, et de l'importance d'examiner les questions au travers de la multiplicité des exemples et des sources. La technique et les connaissances techniques ont un rôle à jouer dans les changements et dans la compréhension de ces changements. Mais c'est au-delà, dans les projets des acteurs et des communautés qu'ils ont réussi à constituer, que l'on peut lire une véritable histoire de l'informatique. Les évolutions techniques sont bien évidemment incapables d'expliquer à elles seules l'impact social de l'informatique et des réseaux. C'est la façon dont des projets, des visions, des utopies se sont concrétisés et ont su rencontrer une demande, un public, des usagers, qui est le ressort principal. Et dans la capacité des informaticiens à donner corps à ces attentes, parfois même à les devancer, comme dans l'approche juridique du Libre, ou la compréhension que le réseau deviendrait ce moyen mondial de communication et à ce titre un objet majeur du changement social. Le logiciel libre, par tout ce qu'il nous dit sur l'informatique d'une part, mais aussi sur le droit, l'économie, la philosophie et la politique, est un excellent objet sur lequel la puissance de l'analyse historique montre sa force.

Introduction générale

Camille PALOQUE-BERGES
Christophe MASUTTI

Le « Libre » avec un L majuscule fait partie de ces termes que l'on sait situer dans un esprit d'époque, mais qui restent complexes à qualifier précisément dans la perspective d'une histoire des idées. Ce qu'on appelle la « philosophie du Libre », au-delà de l'usage parfois flou de l'expression, peut renvoyer à la vision des Lumières des connaissances et de l'action politique : la notion de liberté prend un sens universel en définissant l'attitude intellectuelle et les principes d'action qui concernent toutes les affaires de la cité. Nous préférons ici parler d'éthique du Libre, en soulignant que les principes philosophiques concernant la notion de liberté ont pu s'inscrire dans les usages, mais aussi les technologies, définissant ensemble les conditions d'exercice de ces principes dans des pratiques. À quoi s'appliquent donc aujourd'hui les règles d'action d'une éthique du Libre ? Les champs d'application les plus évidents sont ceux des « logiciels libres » et des « licences libres », expressions aujourd'hui maintenant familières à ceux qui s'intéressent aux mutations sociales et culturelles des pays industrialisés dans les environnements technologiques contemporains. Or, ces mutations en sont venues à tisser un ensemble de correspondances entre des activités techniques, comme la programmation, et l'adhésion à des causes communes comme le partage des œuvres et la

défense des droits face aux enjeux de pouvoir que soulèvent les technologies. Comment passe-t-on ainsi des techniques de programmation aux technologies de la loi ?

Pour comprendre ces correspondances, constitutives d'une « histoire du Libre », il serait insuffisant de ne recourir qu'à l'histoire des idées, elle-même bien souvent victime d'une prétendue linéarité du « progrès technologique ». Un « progrès » qu'elle conçoit tantôt comme une série de ruptures dans l'histoire des hommes, tantôt comme un déterminant, mais dans les deux cas, une boîte noire qu'il faut envisager bien plutôt du côté de l'histoire des techniques. Tout d'abord, parce que c'est au sein des cultures de l'informatique qu'ont commencé à se raconter des histoires du Libre, en particulier celles des pratiques créatives des hackers. Mais il ne s'agit pas seulement d'adopter le point de vue des acteurs, ou de réduire le Libre à une histoire des techniques informatiques. Nous devons réfléchir au Libre en termes de régime technologique, qui se différencie du régime de la technique (dans son rapport raisonné à l'action) en tant qu'il considère l'action non pas en soi, mais en tant qu'environnement ¹. Le régime technologique façonne les objets techniques selon des normes d'usage plus que de prescription. Issu des sciences de la gestion appliquées à l'industrie, ce modèle pourrait expliquer pourquoi et comment des règles d'actions et des modes d'opération expérimentés hors des industries, voire contre elles, finissent par poser des questions d'organisation. C'est ainsi que l'on peut comprendre que le Libre, des logiciels aux licences, se veut bon candidat à la réorganisation des valeurs sociales, culturelles, économiques voire politique dans une société marquée par les technologies.

Notre approche historique se veut donc inspirée par l'histoire des techniques. Tentons cependant de préciser notre positionnement historiographique, puisque nous nous aventurons sur des terrains encore relativement peu balisés par les historiens français. L'apparente contemporanéité des ordinateurs modernes, telle que définie par l'historien Paul Ceruzzi ², se dessine comme un trompe-l'œil lorsqu'on tire les fils historiques qui composent le domaine de l'histoire des machines des années 1950 à nos jours.

1. Anne-Françoise GARÇON, *L'imaginaire et la pensée technique. Une approche historique, XVI^e-XX^e siècle*, Paris : Garnier, 2012.

2. Paul E. CERUZZI, *A History of Modern Computing*, Cambridge, Mass. : MIT Press, 1998.

En effet, l'inventaire des grandes dates de l'informatique, tout comme les grandes étapes de l'innovation (comme les semi-conducteurs ou l'émergence des nouveaux protocoles des réseaux de communication) s'avèrent être autant d'approches insuffisantes si l'on n'y adjoint pas une analyse des pratiques, des usages et des transferts de connaissance. C'est en somme l'objet du plaidoyer récent de Janet Abbate¹ lorsqu'elle montre combien est importante une approche STS (*Science and Technology Studies*) de l'histoire d'Internet.

Comment, justement, raconte-t-on l'histoire de l'informatique aujourd'hui ? L'histoire des machines, de la mécanisation à la computation dans une « physique du calcul »² croise celle des technologies industrielles et l'entrée de l'ordinateur dans le domaine des techniques de la productivité³. On l'aborde également sous l'angle d'une analyse des contextes scientifico-idéologiques qui ont favorisé son essor⁴. Enfin, les nouvelles possibilités en termes de capacités médiatiques et créatives offertes par les interfaces des micro-ordinateurs ont été décrites comme de nouveaux langages⁵, et les réseaux numériques analysés comme des environnements où se redéfinissent aussi bien les logiciels que leurs usages, en co-évolution⁶.

Les programmes aussi ont leurs histoires. Non seulement les lignes de codes sont les fruits de l'ingénierie logicielle, que l'on ne saurait séparer des dynamiques organisationnelles de l'activité de programmation, mais elles sont aussi la traduction technologique qui mêle le besoin de calcul et les déterminants techniques que constituent les machines elles-mêmes. Les grandes lignes de l'historiographie de l'informatique se concentrent sur les machines, les concepts, les idées, l'économie, et les grands hommes. Faire une histoire des logiciels supposerait de prendre

1. Janet ABBATE, « L'histoire de l'Internet au prisme des STS », dans : *Le Temps des médias* 18.1 (2012), p. 170, URL : <http://www.cairn.info/revue-le-temps-des-medias-2012-1-p-170.htm>.

2. Girolamo RAMUNNI, *La physique du calcul. Histoire de l'ordinateur*, Paris : Hachette, 1989.

3. James W. CORTADA, « Researching the History of Software From the 1960s », dans : *IEEE Annals of the History of Computing* 24.1 (2002), p. 72–79.

4. Paul N. EDWARDS, *The Closed World : Computers and the Politics of Discourse in Cold War America*, Cambridge : MIT Press, 1997.

5. Lev MANOVICH, *Le langage des nouveaux médias*, Paris : Les Presses du Réel, 2010.

6. ABBATE, *op. cit.*

cela en charge tout en ouvrant la possibilité d'étudier aussi les phénomènes intermédiaires à ces grandes lignes historiographiques. Ainsi, entre les idées et les concepts, le croisement des discours et des pratiques sur les techniques logicielles ouvre la voie à des imaginaires qui accompagnent leur développement. Parmi les grands noms de l'informatique se trouvent différents agrégats sociaux qui prennent en charge les programmes et se les approprient. De manière similaire, les modèles économiques des industries informatiques sont loin d'être limités à ceux des entreprises, mais rencontrent des logiques d'échanges communautaires qui viennent les transformer. Ainsi, cet ouvrage s'intéresse à l'histoire des techniques logicielles dans leurs trajectoires, c'est-à-dire une évolution qui n'est pas seulement celle d'un progrès linéaire mais qui est marquée par des transformations au gré de différentes logiques de conception, d'utilisation et de représentations qui se rencontrent dans des pratiques collectives.

Le logiciel libre constitue l'une de ces trajectoires, une piste que l'on retrace dès la fin des années 1950, à un moment où se cristallisa un ensemble de pratiques d'écriture et de médiation du code informatique, formalisées pour la première fois au début des années 1980 par Richard Stallman. Il fut le premier à définir et à revendiquer le concept de logiciel libre au travers des libertés assurées à l'utilisateur des programmes. Cette définition qui a fait date n'aurait probablement pas eu autant d'impact si elle n'avait été accompagnée d'un support juridique et organisationnel : la rédaction, avec Eben Moglen, de la licence publique générale GNU, première pierre dans l'ensemble de préconisations défendues par la Free Software Foundation. Plus qu'une simple alternative, le logiciel libre représente un courant parallèle qui diffuse autant qu'il reçoit des autres courants. C'est sur cette base qu'il a d'abord été en opposition avec certaines pratiques et évolutions du monde logiciel comme dans les autres sphères culturelles (musique, cinéma, édition...) : logiciels propriétaires, verrous numériques, limitation de copies, surveillance des réseaux, etc. Les épisodes de cette « révolution du logiciel libre »¹ ont été largement diffusés et commentés, dans ses réappropriations multiples. À des phénomènes de périodisation historique se superposent des positionnements perçus comme idéologiques : le Libre stallmanien naît dans les an-

1. Sam WILLIAMS, Richard STALLMAN et Christophe MASUTTI, *Richard Stallman et la révolution du logiciel libre. Une biographie autorisée*, Paris : Eyrolles, 2010.

nées 1980, il est marqué par une très forte revendication éthique fondée sur la liberté d'expression (les codes informatiques libres étant considérés comme autant d'expressions politiques de programmeurs) et précisant les différences entre liberté et gratuité des logiciels¹ ; les années 1990, elles, voient se rencontrer dans l'*open source* (le code source « ouvert ») les principes de partage des codes issus du Libre et des modèles d'organisation économiques appliqués aux systèmes d'exploitation (via les travaux des collectifs agrégés autour du système d'exploitation Linux) : c'est le modèle torvaldien ou raymondien² du Libre, qui a fortement contribué à populariser le phénomène, notamment grâce à l'idée qu'il pouvait générer une économie profitable. Nous souhaitons dépasser cette opposition, dont l'intérêt est toujours relatif aux enjeux idéologiques (ou parfois seulement rhétoriques) soulevés par les acteurs du Libre. Pour citer Wikipédia : « Les désignations *free software* et *open source* sont en réalité deux désignations concurrentes pour un même type de licence de logiciel. En utilisant la désignation *free software*, on tient à mettre en avant la finalité philosophique et politique de la licence, tandis que la désignation *open source* met l'accent sur la méthode de développement et de diffusion du logiciel. »

Plus encore, le contexte historique de cette « révolution » ne peut être indépendant, en amont, d'un héritage conceptuel qui a rendu possibles les pratiques de partage et, en aval, d'un terrain social, politique et économique favorable à la réception de ces pratiques et de leurs différentes formalisations (licences libres, émergence de communautés libristes, transferts de technologies...). Par ailleurs, ce contexte doit également être décrit dans les termes d'un environnement socio-technique qui aurait non pas déterminé mais favorisé la formulation d'un certain nombre de positionnements face à la conception et à l'usage d'objets techniques de plus en plus inscrits dans notre quotidien. Ainsi, le rôle des réseaux de communication d'Internet n'est pas négligeable dans l'apparition de discours sur les libertés logicielles. En effet, sont inscrits dans les codes mêmes d'Internet (des protocoles aux applications) un certain nombre de principes d'ouverture des techniques pour mieux faire circuler l'information.

1. Rappelons qu'en anglais, le terme *free* veut dire à la fois libre et gratuit, d'où la célèbre précision faite par Stallman : « free as in free speech, not as in free beer ! ».

2. Des noms Linus Torvald et Eric S. Raymond, deux des grandes figures de l'*open source*, respectivement programmeur du noyau Linux et co-fondateur de l'Open Source Initiative.

En deçà des idéologies libertariennes, l'encouragement à la manipulation et l'ouverture des techniques de l'information par et à travers les réseaux favorisent l'idée qu'une mainmise totale sur les artefacts informatiques, et donc leur contrôle, n'est pas possible. D'une redéfinition des gestes de la technique à une redéfinition des normes sociales, juridiques, politiques et culturelles des œuvres et des pratiques, il n'y a qu'un pas.

Les libertés logicielles constituent un ensemble de permissions qui déterminent les conditions d'usage d'un logiciel. Ces conditions sont précisées dans une licence qui accompagne ce dernier. Il y a plusieurs types de licences libres (licence BSD, GNU GPL, CeCILL, etc), répondant à des objectifs convergents mais différents, toutes conformes aux législations nationales et traités internationaux établissant le cadre commun de la propriété intellectuelle. Les licences libres constituent des variations de quatre libertés logicielles fondamentales : liberté d'utiliser le programme, liberté de le copier, liberté d'en étudier le code source, et liberté de le modifier et de diffuser les versions modifiées. Peu à peu, ces libertés ont dépassé le cadre du logiciel pour s'appliquer plus généralement à d'autres types d'informations, notamment sous forme numérique. Il s'agit d'abord des manuels de logiciel libres, et, dans un mouvement plus récent, les données culturelles (musique, littérature, œuvres d'art...) et scientifiques (brevets, publications) soumises au droit d'auteur. Pour autant, les licences libres ne sont pas qu'une simple réaction libertaire et contradictoire face aux régimes traditionnels de propriété intellectuelle (que l'on pourrait exprimer par l'opposition *copyright/copyleft*) ; au contraire, elles suivent le principe d'une définition *a priori* des conditions d'utilisation de leurs œuvres, à ceci près que les acteurs engagés dans cette définition et les utilisations ainsi projetées dépassent largement le cadre habituel de la logique des ayants droits. La pertinence et l'impact de telles licences sur l'innovation et la créativité grâce à la diffusion des œuvres numériques par Internet ont été exposés et discutés par Lawrence Lessig¹ et Philippe Aigrain².

On comprend aisément que ce qui devait se circonscrire comme une trajectoire particulière d'une classe tout aussi particulière de logiciels, s'est

1. Lawrence LESSIG, *L'avenir des idées : le sort des biens communs à l'heure des réseaux numériques*, Lyon : Presses Universitaires de Lyon, 2005.

2. Philippe AIGRAIN, *Internet et Création. Comment reconnaître les échanges hors-marché sur internet en finançant et rémunérant la création ?*, Paris : In Libro Veritas, 2008.

traduit par un élan général qui dépasse des manières d'envisager à la fois le geste technique de la programmation et l'économie des programmes ainsi conçus. En effet, vu comme un modèle, le Libre transcende les secteurs où il est censé s'appliquer et devient culturel, c'est-à-dire véhiculant un ensemble de valeurs et de comportements. Ainsi, pour résumer, le Libre traduit donc l'ensemble des pratiques de programmation et, plus généralement, celles de l'ingénierie du logiciel libre, les principes de partage de connaissances et de biens (culturels, artistiques, techniques, etc.), les modèles d'organisation et d'économie basés sur ces principes, les valeurs culturelles et sociales ainsi que les comportements résultants de l'adhésion à ces modèles et ces valeurs (dont la formalisation se trouve dans les licences libres, inhérentes au droit et à l'éthique).

L'intention de cet ouvrage (y a-t-il des ouvrages innocents ?) n'est donc pas seulement de diffuser un ensemble d'analyses inédites sur les logiciels libres ou les hackers, ces programmeurs de génie, présents depuis la fin des années 1960¹. Cela a longuement été décrit à la fois par des journalistes comme Steven Lévy², des historiens³, la presse spécialisée et même de célèbres hackers par leurs biographies ou auto-biographies⁴. L'intention de cet ouvrage est justement de montrer à quel point l'informatique libre et le Libre en général sont arrivés à ce point de maturité que leurs histoires, au pluriel, ne sauraient se contenter des références unilatérales au monde hacker souvent réduit à un folklore, voire à une caricature. Il y a au contraire plusieurs cultures du Libre et, donc, plusieurs histoires, selon l'approche que l'on adopte pour en définir les contours. Les modèles économiques du Libre ainsi que les pratiques informatiques libristes sont des approches que nous pouvons qualifier de classiques, parce qu'elles ont déjà été tentées avec plus ou moins de succès, souvent au détour d'autres approches plus générales de l'histoire de l'informatique. Les représentations sociales, les idéologies, les études de cas sur des trajectoires concrètes de logiciels libres, la place du Libre dans les institutions

1. Voir en annexe de la biographie autorisée de Richard M. Stallman : « À propos du terme hacker ». WILLIAMS, STALLMAN et MASUTTI, *op. cit.*, p. 289-297.

2. Steven LEVY, *Hackers. Heroes of the Computer Revolution*, New York : Dell Publishing, 1994.

3. Christopher KELTY, *Two bits : the cultural significance of free software*, Durham : Duke University Press, 2008.

4. Linus TORVALD et David DIAMOND, *Il était une fois Linux : L'extraordinaire histoire d'une révolution accidentelle*, Paris : Osman Eyrolles Multimédia, 2001.

et dans le droit, toutes ces approches sont en revanche autant de champs qu'il est urgent de dégrossir.

Notes à propos de l'ouvrage

Sur *framabook.org*, le lecteur pourra trouver d'autres textes complémentaires à cet ouvrage. Rédigés sous forme de tribunes, dont l'apport est sincère et pertinent, leur intégration dans le livre aurait sous-entendu une interprétation trop militante des histoires qui suivent. Si l'éditeur Framasoft milite depuis longtemps pour la promotion et la diffusion du logiciel libre et la culture libre, l'ouvrage ici présent cherche à entretenir un certain recul nécessaire et méthodologique. Pour autant, bien des écueils ont menacé cet ouvrage et nous ne prétendons pas les avoir tous évités. Le premier d'entre eux aurait été une vision finaliste, absolument contraire à notre démarche.

Références

- ABBATE, Janet, « L'histoire de l'Internet au prisme des STS », dans : *Le Temps des médias* 18.1 (2012), p. 170, URL : <http://www.cairn.info/revue-le-temps-des-medias-2012-1-p-170.htm>.
- AIGRAIN, Philippe, *Internet et Création. Comment reconnaître les échanges hors-marché sur internet en finançant et rémunérant la création ?*, Paris : In Libro Veritas, 2008.
- CERUZZI, Paul E., *A History of Modern Computing*, Cambridge, Mass. : MIT Press, 1998.
- CORTADA, James W., « Researching the History of Software From the 1960s », dans : *IEEE Annals of the History of Computing* 24.1 (2002), p. 72–79.
- EDWARDS, Paul N., *The Closed World : Computers and the Politics of Discourse in Cold War America*, Cambridge : MIT Press, 1997.
- GARÇON, Anne-Françoise, *L'imaginaire et la pensée technique. Une approche historique, XVI^e-XX^e siècle*, Paris : Garnier, 2012.
- KELTY, Christopher, *Two bits : the cultural significance of free software*, Durham : Duke University Press, 2008.

- LESSIG, Lawrence, *L'avenir des idées : le sort des biens communs à l'heure des réseaux numériques*, Lyon : Presses Universitaires de Lyon, 2005.
- LEVY, Steven, *Hackers. Heroes of the Computer Revolution*, New York : Dell Publishing, 1994.
- MANOVICH, Lev, *Le langage des nouveaux médias*, Paris : Les Presses du Réel, 2010.
- RAMUNNI, Girolamo, *La physique du calcul. Histoire de l'ordinateur*, Paris : Hachette, 1989.
- TORVALD, Linus et David DIAMOND, *Il était une fois Linux : L'extraordinaire histoire d'une révolution accidentelle*, Paris : Osman Eyrolles Multimédia, 2001.
- WILLIAMS, Sam, Richard STALLMAN et Christophe MASUTTI, *Richard Stallman et la révolution du logiciel libre. Une biographie autorisée*, Paris : Eyrolles, 2010.

Remerciements

Nous remercions tout d'abord les auteurs qui, depuis l'envoi de leurs propositions de chapitres au printemps 2012 jusqu'aux dernières corrections sur les textes définitifs en hiver 2013 ont bien voulu travailler à la constitution d'un ouvrage que l'on espère cohérent.

Le travail sur les textes n'aurait pas été possible sans l'accompagnement bénévole de l'équipe des correcteurs et de contributeurs au projet Framabook, dont les relectures attentives ont été précieuses à plus d'un titre. Par avance, nous remercions de même tous les lecteurs qui voudront nous faire part de leurs remarques, commentaires et suggestions d'amélioration en participant à ce livre libre depuis le site <http://framabook.org>.

I

(Pré-) histoire

Des réseaux ouverts, pour quoi faire ?

Valérie SCHAFER

Nous croyons qu'un Internet libre et ouvert peut apporter un monde meilleur. Pour conserver l'Internet libre et ouvert, nous appelons les communautés, industries et pays à reconnaître ces principes.¹

Ainsi s'ouvre le Préambule de la *Declaration of Internet Freedom* de juillet 2012, appel international signé par des organisations comme l'American Civil Liberties Union, Amnesty International, Free Press, l'Electronic Frontier Fondation et des grands noms de l'Internet, pères fondateurs, théoriciens des communs, chercheurs, ou encore dissidents et activistes célèbres, parmi lesquels Ai Weiwei, John Perry Barlow, Yochai Benkler, danah boyd ou Vinton Cerf, pour n'en citer que quelques-uns.

Interconnexion de plus de 50 000 réseaux, Internet incarne en effet le modèle du réseau ouvert. En plaçant « l'intelligence » à l'extrémité du réseau (bout en bout), en permettant à des réseaux de nature différente de s'interconnecter, le « réseau des réseaux » a fondé sa croissance et sa réussite sur l'ouverture.

Celle-ci n'est pas seulement technique, elle est également organisationnelle : le choix d'un mode de discussion ouvert des protocoles, qui prend

1. <http://www.internetdeclaration.org/freedom>. Nous traduisons.

la forme de *Request For Comments*¹ (et ce dès ARPANET), la création de l'*International Network Working Group*, puis de l'*Internet Engineering Task Force* et de l'*Internet Society*, ou encore la constitution du *World Wide Web Consortium* pour le développement du Web dans la première moitié des années 1990, constituent des ruptures par rapport aux cadres de normalisation historiques (l'*International Organization for Standardization* notamment) et aux standards *de facto* des entreprises privées (architecture SNA d'IBM par exemple).

La gouvernance de l'Internet se bâtit peu à peu sur des bases nouvelles, au sein d'une « République des informaticiens »² qui met en œuvre un modèle organisationnel inédit, qui semble davantage apparenter le réseau à un « bien commun ».

En parallèle, un autre projet porte dans les années 1980 et au début des années 1990 les promesses et les valeurs de l'ouverture : il s'agit de l'*Open Systems Interconnection* (OSI), une architecture en sept couches développée au sein de l'*International Organization for Standardization* (ISO). Or, Internet s'impose dans les années 1990 au détriment de l'OSI, processus certes œcuménique mais aussi plus lent et coûteux.

Leurs trajectoires parallèles sont intéressantes pour interroger l'ouverture à l'œuvre dans les réseaux de données à partir d'ARPANET, à la fois sous l'angle des protocoles et de leur mise en œuvre, mais aussi des enjeux de gouvernance, des imaginaires et des discours qui leur sont attachés.

L'ouverture, que ce soit dans le domaine technique ou organisationnel, n'est pas sans susciter des controverses, qui se sont accélérées avec le passage de l'Internet dans le grand public durant les années 1990/2000, ce qu'illustrent notamment les débats houleux autour de la Neutralité de l'Internet ou de la gouvernance.

Aussi, ce chapitre souhaite également interroger les motivations qui fondent la notion d'ouverture, très diverses et finalement solidement ancrées dans les contextes économiques et politiques de leur temps, et les mettre en parallèle avec les enjeux actuels et les évolutions qu'a subi le

1. Documents de spécifications techniques ouverts, décrivant les protocoles de l'Internet.

2. Patrice FLICHY, « Internet ou la communauté scientifique idéale », dans : *Réseaux* 17.97 (1999), p. 77–120, p. 90.

modèle Internet sous l'effet de la popularisation du Web, du développement de modèles clients/serveurs qui l'éloignent de l'architecture horizontale originelle ou de l'introduction de mesures destinées à garantir la qualité de service. La notion d'ouverture s'est en effet peu à peu teintée de nuances qui servent des discours et des projets aux valeurs parfois franchement antagoniques.

L'ouverture dont il est question n'est bien sûr pas celle des communautés *open source*, mais en invitant à un léger décalage au sein de l'histoire de l'informatique, ce chapitre cherche à travers la question des architectures ouvertes de réseaux, à rappeler un certain nombre de motivations et de principes fondateurs qui traversent des communautés pionnières, qui ne sont pas étanches les unes aux autres comme le montre la rencontre au début des années 1980 d'Unix et TCP/IP. Le caractère collaboratif d'élaboration des protocoles, le refus des systèmes propriétaires se font écho au sein des différentes communautés, de même que les implications techniques, politiques, économiques ou idéologiques sous-tendues par ces questions.

1. Les réseaux seront ouverts ou ne seront pas

Lorsque le projet ARPANET est lancé dans la seconde moitié des années 1960 par l'IPTO (Information Processing Techniques Office), au sein de l'Advanced Research Projects Agency, agence militaire américaine, l'ouverture et l'hétérogénéité dans le domaine des réseaux sont tout sauf évidentes.

En avril 1967, personne n'a la moindre idée de la façon de faire communiquer des ordinateurs hétérogènes. Les langages de programmation, les systèmes d'exploitation, les ordinateurs... : tout sépare le Sigma-7 de UCLA du TX-2 du MIT ou du SDS-940 du SRI (pour ne prendre que ces exemples). Chaque ordinateur est alors un « îlot » technique, totalement incompatible avec ses semblables. Les faire communiquer entre eux relève déjà de la gageure, mais vouloir le faire selon les plans présentés par Roberts va apparaître, aux plus avertis comme Wes Clark, comme une insanité technique.¹

1. Alexandre SERRES, « Aux sources d'internet : l'émergence d'ARPANET », Thèse en Sciences de l'information et de la communication, Rennes : Université de Rennes, 2000, p. 415.

Il convient de rappeler les réticences émises lors de la réunion annuelle de l'IPTO d'Ann Arbor en avril 1967, au cours de laquelle Robert Taylor et Larry Roberts présentent les concepts fondateurs d'ARPANET aux chercheurs qui, dans les universités liées à l'ARPA, sont censés prendre part au projet. Les réactions vont de l'indifférence à l'hostilité : certains s'inquiètent de devoir partager leurs ressources avec d'autres centres et ne souhaitent pas perdre le contrôle local sur leurs machines, d'autres ont le sentiment que c'est un argument en faveur d'une moindre dotation en équipements informatiques¹.

Toutefois, ce choix décisif qui trouve sa source dans ARPANET va non seulement lever des problèmes d'incompatibilité et d'interopérabilité, mais aussi jeter les bases qui vont garantir la croissance de l'Internet.

1.1 L'hétérogénéité, une rupture engagée par ARPANET

La mention de Larry Roberts lors de la réunion d'Ann Harbor n'est pas anecdotique. En effet, bien que l'idée de faire communiquer des matériels hétérogènes puisse sembler incongrue à une majorité de chercheurs en 1967, Larry Roberts, lui, a déjà affronté le problème. En 1965, il a réussi avec Thomas Marill une connexion à grande distance entre deux ordinateurs de types différents, un TX-2 au Lincoln Laboratory et un Q-32 d'IBM à Santa Monica. Il ne s'agit pas à proprement parler d'un réseau, mais d'un premier pas vers ce qui devient ensuite, dans ARPANET, un réseau hétérogène. Leurs résultats sont publiés juste avant que Larry Roberts passe à l'ARPA, où ces travaux sont suivis de près depuis l'IPTO par son directeur, Ivan Sutherland et Robert Taylor, son adjoint.

Ce dernier devient directeur de l'IPTO en 1966 et lance ARPANET, un projet encore vague, mais qu'il conçoit déjà comme distribué et hétérogène.

À l'origine de cette vision on mentionne souvent la présence à l'IPTO de trois terminaux connectés chacun à trois sites importants liés à l'ARPA : ils permettent de suivre à distance les travaux effectués, voire d'envoyer des formes rudimentaires de messages électroniques, mais chacun est installé sur une ligne dédiée et ils sont incompatibles. C'est un problème très

1. *ibid.*, pp. 411-413. Janet ABBATE, *Inventing the Internet*, Cambridge, MA : MIT Press, 1999, p. 50.

concret, qui motive l'ouverture... et qui n'est pas sans rappeler les débuts de Richard Stallman face à son imprimante Xerox¹.

Toutefois, comme le souligne Alexandre Serres : « Certes les procédures de connexion étaient longues, compliquées et fastidieuses et l'incompatibilité du matériel comme les limites de cet embryon de réseau technique ont certainement poussé Taylor à rechercher une autre solution de communication. De là à en faire l'élément *déclencheur* d'ARPANET, il y a un pas que nous refusons de franchir : nous avons trop insisté sur la multiplicité des origines d'ARPANET pour réduire tout à coup l'explication de la décision de Taylor à ce seul facteur technique. »²

En 1966, Larry Roberts prend en charge la construction du réseau informatique et inscrit dans les principes d'ARPANET de mettre fin aux structures centralisées des réseaux, en interconnectant une grande variété d'ordinateurs présents sur les sites universitaires liés à l'ARPA (machines d'IBM, DEC, General Electric, Univac...).

La conceptualisation commence en 1967. Y travaillent des pionniers dans le domaine des réseaux comme Leonard Kleinrock, d'anciens camarades de Larry Roberts à UCLA, quelques étudiants comme Vinton Cerf et Jon Postel. Robert Kahn rejoint l'aventure en 1968, par le biais de BBN (Bolt, Beranek and Newman), firme de consultants qui obtient le contrat avec l'ARPA pour créer les IMP (Interface Message Processors), qui forment un sous-réseau dans ARPANET permettant la communication entre ordinateurs hétérogènes.

Le réseau démarre en 1969 et dispose de quatre connexions entre universités, la première à UCLA, puis au Stanford Research Institute, à UCSB (Santa Barbara) et à l'Université d'Utah. En 1972, la conférence internationale sur les communications informatiques voit la première démonstration publique d'ARPANET.

Alors que son protocole, NCP, propose une solution pour connecter des matériels hétérogènes, la multiplication de réseaux de nature différente

1. Sam WILLIAMS, Richard STALLMAN et Christophe MASUTTI, *Richard Stallman et la révolution du logiciel libre. Une biographie autorisée*, Paris : Eyrolles, 2010, chap. 1.

2. SERRES, *op. cit.*, p. 397.

contribue à élargir le problème et engager Vinton Cerf et Robert Kahn dans la recherche d'une alternative¹.»

En 1974, les deux chercheurs publient un papier qui définit le protocole TCP (*Transport Control Protocol*), qui évolue en TCP/IP, protocole clé de l'Internet qui permet à différents réseaux de communiquer (radio, satellites, locaux, etc.) et assure les bases de l'Internet en pariant sur la commutation de paquets et les datagrammes², le bout en bout, l'interopérabilité et l'ouverture.

L'idée de relier des réseaux, et non plus « simplement » des matériels hétérogènes, a également donné lieu à des propositions de la part du français Louis Pouzin, connues sous le terme *catenet*³, avant que le mot Internet ne s'impose.

1.2 Les motivations du réseau ouvert Cyclades

Louis Pouzin est alors responsable à l'Institut de Recherche en Informatique et Automatique (IRIA) du projet Cyclades, qui se fonde sur des principes proches de ceux d'ARPANET.

Cyclades réseau démarre en 1971 et s'éteint en 1979, malgré des apports techniques, dont certains seront intégrés par l'équipe ARPANET, en particulier les principes de *windowing*⁴ et de datagrammes.

À l'exception du réseau ARPANET, alors en cours de constitution, la tendance est davantage aux architectures propriétaires bâties par des constructeurs qui disposent ainsi d'un marché captif lié à leurs équipements. Les motivations qui poussent Louis Pouzin et son équipe vers une architecture technique ouverte sont intéressantes : le contexte politique et économique a un rôle décisif.

1. Archives orales du Charles Babbage Institute, Interview de V. Cerf par Judy O'Neill, 24 avril 1990, <http://www.cbi.umn.edu/oh/pdf.phtml?id=81>.

2. La commutation de paquets consiste à découper un message en paquets avant de les faire circuler dans le réseau et de les rassembler à destination. Quand on utilise un mode datagramme, les paquets circulent dans le réseau selon un routage adaptatif : ils peuvent suivre des chemins différents.

3. Louis POUZIN, « A Proposal for Interconnecting Packet Switching Networks », dans : *Proceedings of EUROCOMP*, Bronel University, 1974, p. 1023–1036.

4. Le principe des fenêtres ou *windowing* concerne la numérotation des paquets introduits dans le réseau. Cette fenêtre plus ou moins longue incorpore les éléments de numérotation des paquets et des accusés de réception qui circulent.

En effet, l'IRIA a été créé en 1966 dans le cadre du Plan Calcul, voulu par le Général de Gaulle pour soutenir la politique informatique française et s'émanciper de la domination américaine. L'IRIA est la branche recherche de cette politique, qui a aussi une dimension industrielle avec la création de la CII (Compagnie Internationale pour l'Informatique) en 1966.

L'idée de Louis Pouzin et de son équipe repose sur un certain pragmatisme : le réseau Cyclades doit avoir des liens avec les matériels de la CII, mais les futurs réseaux ne seront pas constitués exclusivement d'équipements CII, sachant qu'IBM occupe une place dominante sur le marché mondial.

IBM a développé en 1974 une architecture propriétaire, SNA, qui limite l'extension d'un réseau à ses propres calculateurs. En démontrant qu'un réseau peut accueillir des machines de constructeurs différents, l'équipe Cyclades peut envisager de faire de la place à la CII dans le domaine des réseaux.

À partir du moment où l'on voulait promouvoir la CII, il fallait faire de l'hétérogène¹ car il n'y avait pas de monopole, il fallait bien admettre qu'il n'y aurait pas que la CII dans un réseau, qu'il y aurait forcément un certain nombre de machines IBM, parce que ce sont eux qui mangeaient le marché, et il fallait bien admettre aussi que des gens ont des Control Data, des Burroughs. [...]

— VS : Donc c'était une idée plutôt pragmatique ?

— LP : Qui était non seulement pragmatique mais considérée comme politiquement correcte, cela permettait de faire rentrer un coin dans les systèmes IBM, en disant : « Prenez une techno style Cyclades ou ARPA, et comme cela vous pourrez connecter votre machine, vous ne serez pas obligés de prendre de l'IBM ». ²

Juste avant d'intégrer le projet Cyclades, Louis Pouzin a travaillé chez Simca à Poissy. C'est sa première expérience concrète en matière de lutte contre la « dominance » d'IBM et de réflexion sur les systèmes propriétaires : « IBM faisait la loi, disait ce qu'il fallait acheter, etc. C'était un

1. Un réseau homogène est constitué des machines d'un même constructeur, un réseau hétérogène incorpore des machines de constructeurs différents, qui utilisent des langages de programmation et/ou des systèmes d'exploitation différents.

2. Entretien de Valérie Schafer avec Louis Pouzin du 12 novembre 2002.

peu le style aussi aux États-Unis, le patron de l'informatique aux États-Unis auquel on devait envoyer les rapports était aussi un fanatique d'IBM. Alors je me suis amusé par exemple à remplacer les disques IBM par des compatibles, ce qui a jeté pas mal de troubles. »¹

Louis Pouzin développe notamment un petit système d'accès télématique, mais au lieu de s'appuyer sur des terminaux IBM, il choisit des télétypes, et préfère au logiciel IBM un petit système développé par des collègues. L'ouverture, tant dans le domaine matériel que logiciel, l'attire déjà. Il va pouvoir reprendre dans Cyclades ses expériences en matière d'hétérogénéité et de lutte contre la domination d'IBM, mais cette fois en matière de réseaux. Si Christopher Kelty note : « Il y a bien un héritage commun à UNIX et l'ARPANET, via la figure de Joseph Carl Robnett Licklider, impliquant l'imaginaire des systèmes d'exploitation à temps partagé couplé au rêve du « réseau intergalactique » ; mais les développements d'ARPANET et d'UNIX ont d'abord été complètement indépendants. UNIX permettait le partage des ressources sur un seul ordinateur (selon le principe du temps-partagé), qu'il soit ordinateur central ou un mini-ordinateur, mais n'était pas destiné à être connecté à un réseau – c'est toujours le cas aujourd'hui. À l'opposé, le but de l'ARPANET était explicite : permettre le partage des ressources à partir de machines à distance sur différents réseaux »², on voit ici que le thème de la compatibilité ou de l'ouverture peut se déplacer au fil de la carrière d'un individu, tout en gardant les mêmes soubassements idéologiques.

On observe également l'imbrication dans les choix techniques du contexte politico-économique et de l'initiative personnelle : Louis Pouzin n'a pas attendu Cyclades pour s'intéresser aux systèmes ouverts, il commence chez Simca, mais sa volonté rencontre celle de la Délégation à l'Informatique dans les années 1970... et heurte de plein fouet un autre monopole, national, celui des télécommunications.

Sans développer le conflit en France entre la solution préconisée par Cyclades (les datagrammes) et celle du monde des télécommunications retenue dans le réseau Transpac, qui a supporté jusqu'en juin 2012 le tra-

1. Entretien de Valérie Schafer avec Louis Pouzin du 12 novembre 2002.

2. Ce passage est extrait de la traduction par Camille Paloque-Bergès. Christopher KELTY, *Two bits : the cultural significance of free software*, Durham : Duke University Press, 2008, p. 118-143.

fic Minitel (choix des circuits virtuels¹), cette opposition n'est pas sans conséquence sur la fin de Cyclades et son extinction en 1979.

1.3 Internet : l'ouverture garante de la croissance et de l'innovation

Alors que Cyclades s'éteint, la décennie 1980 est féconde en évolutions dans le monde informatique et plusieurs éléments vont contribuer à la croissance d'Internet, dont deux nous intéressent particulièrement : il s'agit d'une part du développement par la *National Science Foundation* de NSFNET au milieu des années 1980, réseau de la recherche en science qui se fonde à partir de 1986 sur TCP/IP, et ouvre l'Internet à la presque totalité des universités américaines, et d'autre part de la rencontre avec Unix, système d'exploitation dont les principaux atouts sont la « simplicité », la portabilité sur plusieurs types de machines et le prix.

Unix est né dans l'esprit de chercheurs d'AT&T, Dennis Ritchie et Ken Thompson, qui travaillaient au développement d'un système d'exploitation multi-tâches pour les mini-ordinateurs apparus notamment chez Digital Equipment Corporation. À la suite de la décision de la Justice américaine d'empêcher AT&T d'en tirer profit en vertu des lois anti-trust, il est offert à bas prix ou donné aux universités. Ainsi Unix est rapidement adopté par les universités américaines. L'ARPA convainc l'Université de Berkeley de modifier le système pour intégrer ses protocoles et la nouvelle version d'Unix en 1983 est dotée de TCP/IP.

Contrairement au mythe d'une fondation très libérale et libertarienne de l'Internet, on doit insister ici sur le poids de l'ARPA ou de la NSF, et donc des financements gouvernementaux, dans le déploiement de l'Internet. Comme le note Paul Ceruzzi :

En 1986, la NSF relia cinq centres de superordinateurs, et prit trois décisions cruciales pour la suite de l'histoire. La première fut d'adopter les protocoles TCP/IP ; *a posteriori*, cette décision semble évidente ; elle ne l'était pas à l'époque. La seconde fut de créer un réseau général, disponible pour l'ensemble des chercheurs. La troisième fut de financer la construction d'une « dorsale » à haut débit,

1. Contrairement aux datagrammes, dans le mode circuits virtuels, les paquets n'ont pas un routage adaptatif mais se suivent tous. Valérie SCHAFER, *La France en réseaux*, Paris : Editions Nuvis, 2012.

traversant le pays, à laquelle les réseaux locaux et régionaux pourraient se connecter. En 1987, la NSF lança un appel d'offres pour remplacer la *backbone* d'origine par un nouveau, fonctionnant à une vitesse de 1,5 million de bits par seconde (Mbps), amélioré plus tard à 45 Mbps.¹

TCP/IP s'intègre facilement tant dans NSFNET que dans la version Berkeley 4.2 d'Unix et les utilisateurs passeront, sans même voir leurs usages changer (et notamment l'accès aux *Newsgroups* de la communauté Usenet), par exemple d'UUCP (Unix-to-Unix Control Protocol) à TCP/IP.

Les vertus de TCP/IP résident dans le choix du bout en bout, un pilier fondateur de l'Internet présenté en 1981 par Saltzer, Reed et Clark². Le protocole de l'Internet repousse « l'intelligence » dans les deux bouts de la communication (*end-to-end*) : le serveur et le poste client. Aussi, si un serveur décide d'inventer une nouvelle forme de codage de l'information (par exemple quand Real Networks a créé le premier protocole de streaming audio en 1995), il n'a qu'à diffuser des outils de lecture (les *players* « RealAudio ») et expérimenter son produit. Seul le succès public vient arbitrer entre les services.

David Clark, qui reçoit en 1990 un *Sigcomm Award*³ pour sa contribution à l'Internet, souligne ainsi à quel point l'idée d'ouverture est consubstantielle du « réseau des réseaux » :

Le mot slogan⁴ qui capture le caractère de l'Internet est « ouvert ». L'Internet est ouvert à bien des égards : c'est une plate-forme ouverte (au niveau de la couche Internet) avec des barrières faibles à l'innovation. Les normes sont ouvertes en ce sens qu'elles sont

1. Paul E. CERUZZI, « Aux origines américaines de l'Internet : projets militaires, intérêts commerciaux, désirs de communauté », dans : *Le Temps des Médias 18.1* (2012), p. 15, URL : <http://www.cairn.info/resume.php?IDARTICLE=TDM0180015>.

2. John SALTZER, David REED et David CLARK, « End-to-End Arguments in System Design », dans : *Second International Conference on Distributed Computing Systems*, Versailles, 1981, p. 509–512, URL : <http://web.mit.edu/Saltzer/www/publications/endoend/endoend.pdf>.

3. Ce prix créé en 1989 honore une personnalité du monde informatique ayant mené des travaux significatifs dans le champ des réseaux de données et des communications informatiques.

4. David Clark parle de *one-word bumper-sticker*, expression difficile à traduire en français. En montrant la volonté d'accoler l'adjectif *open* à un certain nombre de mots pour les parer de valeurs positives, comme il l'exprime dans la suite, il pressent l'*openwashing* actuel.

libres de toute barrière IP ou de licences. Les protocoles autorisent *chacun* à devenir un fournisseur de services et rejoindre le club. L'émergence de ces fonctions n'est pas un hasard : les concepteurs de l'Internet les ont valorisées et défendues. Et de nombreux liens ont été revendiqués entre ces caractéristiques et les vertus plus larges de l'*ouvert*, qui est un mot très puissant et positif, qui peut être inséré devant de nombreux noms pour exprimer une valeur positive : l'ouverture des frontières, la conversation ouverte, un visage ouvert, l'accès ouvert, l'ouverture des marchés. *Open* est bon, *fermé* est mauvais. L'Internet, dans une large mesure par sa conception, mais aussi par un heureux hasard, a un caractère technique qui l'aligne sur ce grand ensemble de valeurs qui veulent dire, à l'extrême, le pouvoir de la démocratie et la liberté.¹

Or, comme le souligne Andrew Russell, il y a une ironie cruelle dans le fait que l'on associe aujourd'hui presque exclusivement à Internet l'histoire des réseaux ouverts, en oubliant souvent la tentative concomitante de l'*Open Systems Interconnection* :

La dernière et peut-être la plus cruelle ironie est venue quand les ingénieurs de l'Internet et ses défenseurs ont enfilé le manteau de l'*ouverture* de l'OSI. Pour des raisons évidentes, les ingénieurs de l'Internet dans les années 1980 n'ont pas utilisé le concept pour décrire leur travail. Le premier des documents techniques de l'Internet à attribuer des qualités *ouvertes* aux protocoles de l'Internet et au processus de normalisation a été publié en 1992 par Lyman Chapin, l'un des rares Netheads qui travaillait avec aisance à cheval sur les limites techniques et sociologiques entre OSI et Internet.²

1.4 La guerre des protocoles

Hubert Zimmermann, un des membres de l'équipe Cyclades de Louis Pouzin, a été très actif dans la conceptualisation de ce qui devient l'*Open Systems Interconnection* (OSI)³. Cette architecture de réseau en sept

1. David CLARK, « Network Neutrality : Words of Power and 800-Pound Gorilla », dans : *International Journal of Communication* 1 (2007), p. 701–708, URL : <http://groups.csail.mit.edu/ana/Publications>.

2. Andrew RUSSELL, *An Open World : History, Ideology, and Network Standards*, Cambridge, MA : Cambridge University Press, à paraître.

3. NdE : Acronyme à ne pas confondre avec l'Open Source Initiative !

couches est un cadre de référence couvrant l'ensemble des protocoles de communication qui doivent être mis en œuvre pour l'interconnexion de systèmes hétérogènes.

L'ISO amorce la coordination des travaux en 1977. Mais les réflexions sont engagées depuis 1972 à l'INWG, groupe de réflexion lancé au moment de la première démonstration publique d'ARPANET et rattaché en 1974 à l'International Federation for Information Processing. Ce groupe, dont le président est Vinton Cerf, s'est donné comme objectif de mener des expériences et de proposer des bases de standardisation aux organismes compétents.

Lancé dans l'enthousiasme en 1977, le modèle de référence de base de l'OSI est officiellement publié en 1984, mais ses développements continuent dans les années qui suivent.

L'OSI est un monument avec des développements successifs. [...] L'OSI ne s'est jamais vraiment terminé, c'est tombé en quenouille. [...] Il y a plusieurs raisons : la première est que l'OSI était devenu beaucoup trop compliqué, au niveau de la conception, il y a un vice que les Américains appellent *Bells and Whistles*, « mettre des cloches et des sonnettes pour décorer les choses ». Cela devenait tellement riche et compliqué que cela devenait difficile à saisir et à mettre en œuvre.¹

Parmi les freins que rencontra l'OSI, on peut notamment mentionner une prolifération de normes de fait, en particulier TCP/IP, ou la tentative de normes transitoires (au Royaume-Uni avec les *coloured books* par exemple), et la lenteur et la complexité du processus de normalisation. L'OSI a tendance à devenir une « tour de Babel »². D'autres facteurs vont desservir l'OSI, parmi lesquels son coût, face à un protocole TCP/IP qui s'impose progressivement.

Deux éléments nous intéressent particulièrement dans cette histoire : le premier est la réflexion de fond sur l'ouverture qu'engage l'OSI.

L'OSI suscite en effet des débats sur l'interprétation de la normalisation : faut-il normaliser une « interconnexion des systèmes ouverts » ou une « architecture des systèmes ouverts » ? C'est la première notion qui

1. Entretien de Valérie Schafer avec Marc Levilion du 25 février 2003.

2. ABBATE, *op. cit.*, p. 151.

l'emporte, car il ne s'agit pas de normaliser le comportement interne d'un système d'exploitation mais la visibilité qu'il offre à partir de l'extérieur.

Car le vrai problème n'était pas l'existence même des architectures comme SNA, ou BNA, ou DNA, ou tout autre XNA. . . qui n'étaient que des architectures privées ou *propriétaires* de réseau, conçues pour mettre de l'ordre dans le fonctionnement en réseau des produits de tel ou tel constructeur (et ce désir de mettre de l'ordre dans des lignes de produits hétéroclites privés était non moins légitime que celui d'avoir des *Systèmes ouverts*), le vrai problème était de savoir si ces architectures privées de réseau sont *ouvertes* ou non aux autres. Par conséquent, il fallait définir non pas des normes de réseau pour remplacer des architectures privées existantes, mais plutôt des normes internationales qui rendent ces dernières *ouvertes*. [. . .] Le champ d'application de l'OSI fut ainsi clairement établi pour définir non plus l'architecture d'un système mais pour interconnecter des systèmes ouverts (interconnexion des systèmes ouverts).¹

Les discussions sont âpres sur la définition de l'ouverture, pour savoir si chaque machine doit conserver son propre langage et son architecture sans se préoccuper de la manière dont l'information est échangée ou s'il faut au contraire entrer au cœur du système.

Un second point notable est le faible investissement de l'équipe ARPANET dans la définition de l'OSI, alors qu'elle a initié la réflexion dans le cadre de l'International Network Working Group. Janet Abbate note à ce sujet : « Le fait que les protocoles de l'Internet n'avaient pas été établis par un organisme de normalisation officiel et le fait qu'ils provenaient des États-Unis (qui dominait déjà le marché de l'informatique), rendaient politiquement impossible pour eux le fait d'être accepté par l'ISO comme normes des systèmes ouverts. »²

Elle explique que Vinton Cerf considérait comme très difficile de faire reconnaître son point de vue à l'OSI et qu'il regrettait qu'il manque une couche « internet » dans l'Open Systems Interconnection. Andrew Russell rappelle aussi l'expérience avortée de Vinton Cerf en 1975 auprès du Comité Consultatif International Télégraphique et Téléphonique, organisme de normalisation pour le monde des télécommunications :

1. Échange de courriers électroniques avec Cuong Ngomai en 2003.
2. *ibid.*, p. 173.

Cerf, découragé par sa première excursion dans la diplomatie internationale en matière de normes, a démissionné de son poste de président de l'INWG fin 1975, à peu près en même temps qu'il quittait son poste de professeur adjoint à l'Université de Stanford et qu'il acceptait l'offre de travailler avec Bob Kahn à l'ARPA. Dans les années suivantes, Cerf et Kahn dirigèrent l'élaboration des protocoles de l'Internet dans un environnement qu'ils pourraient contrôler — la petite communauté des contractants de l'ARPA. Le départ de Cerf de l'INWG a marqué un tournant au sein des *Netheads*, une scission entre la communauté Internet et l'alliance internationale de chercheurs. En quelques années, comme nous le verrons, les deux camps sont devenus rivaux.¹

2. Du réseau « libertaire » au réseau « libéral »

L'opposition entre les solutions TCP/IP et OSI vont notamment être virulentes sur le sol européen à la fin des années 1980 et au début des années 1990, autour de la constitution de réseaux de la recherche, que ce soit au niveau national (on voit ainsi s'opposer en France les tenants de la solution IP et de l'OSI au moment de la constitution du réseau RENATER) ou européen (constitution d'Ebone, Europanet...).

Face à une solution normalisée par un organisme reconnu (l'ISO), les tenants de TCP/IP imposèrent la formule du *rough consensus* et *running code* et un mode de normalisation inédit, incarnant une forme de démocratie technique qui n'est pas toutefois sans susciter certaines réserves et critiques.

2.1 Des modes de normalisation en rupture

Dès 1968, pour réfléchir aux premières spécifications d'ARPANET, un NWG (*Network Working Group*) fut mis en place autour de Steve Crocker. Ce dernier définit les *Request for comments* (RFCs), un mode d'échange de documentation et de spécifications techniques ouvert, dédié à la mise au point de normes consensuelles. À ce jour, il y a plus de 6 000 RFC².

1. RUSSELL, *op. cit.*

2. Voir <http://www.ietf.org/rfc.html>.

Outre que les RFCs constituent une mémoire du réseau extraordinaire pour les historiens, cet exemple d'ouverture fondé sur la compétence technique, la discussion plus informelle et pair-à-pair, sachant s'affranchir des hiérarchies universitaires, plus souple et flexible que l'univers des organismes de normalisation traditionnels, va à la fois permettre l'adaptabilité du réseau, mais aussi un processus d'intéressement large des acteurs.

Ce phénomène d'intéressement n'est pas négligeable. L'ouverture garantit aussi la publicité. Comme le note Vinton Cerf :

Un gars nommé Gérard Lelann était à l'IRIA et travaillait avec Pouzin. Il est venu à mon laboratoire à Stanford pendant un an et a eu beaucoup à voir avec les premières discussions sur ce à quoi le protocole TCP pourrait ressembler. Bob Metcalfe, aussi. Metcalfe était à Xerox à l'époque et en juin 1973, nous avons commencé à travailler ensemble, Lelann, Metcalfe et moi, sur la conception du protocole hôte-à-hôte pour Internet. Finalement Metcalfe s'est impatienté de la vitesse à laquelle les choses se passaient. J'ai essayé d'obtenir d'un grand nombre de personnes un accord sur un ensemble de protocoles, et à chaque fois qu'arrivait un nouvel acteur, il fallait repasser par l'argumentaire. Pendant ce temps Metcalfe avait cinq ou six gars chez Xerox qui essayaient de faire fonctionner les réseaux locaux. Finalement, ils ont dit qu'ils ne voulaient pas attendre que ce processus d'entente et de consensus soit conclu, et ils sont partis sur un angle légèrement différent et ont inventé XNS, qui a fait des choix différents de ceux de TCP. Et ils l'ont exécuté avant nous, en fait. Bien sûr, sur le long terme, nous avons... Ils ont gardé le secret et c'était une erreur, je suppose, en regardant aujourd'hui en arrière. S'ils n'avaient pas gardé le secret, nous pourrions tous utiliser XNS au lieu de TCP. Mais en l'état, TCP s'est avéré être le protocole ouvert dans lequel tout le monde a mis la main à un moment ou un autre.¹

Alexandre Serres, dans l'analyse qu'il fait des premiers systèmes à temps partagé, livre des conclusions finalement proches quand il compare les systèmes proposés par Corbató et Teager : selon lui, la postérité de CTSS (*Compatible Time Sharing System*) est liée à la fois à son ouverture technique et à sa capacité d'intéresser et d'enrôler de nombreuses

1. Interview de Vinton Cerf avec Judy O'Neill. Déjà cité. Nous traduisons.

entités sociales et techniques. La relative simplicité du système sera également un atout¹. On retrouve ici des clés du succès de TCP/IP ensuite ou d'Unix (alors que l'OSI comme Multics ont au contraire été jugés trop complexes).

Au plan informationnel, le NWG (Network Working Group) est le premier forum socio-technique à mettre en place une *documentation ouverte* (les RFC) [...].

L'aspect social et culturel est tout aussi essentiel : dans la lignée de celles du *time-sharing* [temps partagé], le NWG est l'une de ces premières communautés en ligne, développant les pratiques propres à la *culture Internet*. C'est le NWG, notamment, qui symbolise le mieux la culture étudiante au sein des acteurs d'ARPANET, imprégnés des idées libertaires des sixties. [...]

Le quatrième domaine enfin où l'action du NWG peut également apparaître importante est le domaine organisationnel : le NWG est la première véritable ébauche spontanée des futures organisations de l'Internet. Né du réseau ou à partir de lui, le NWG est l'exemple même des comités *ad hoc*, qui ne cesseront d'accompagner le développement de l'Internet.²

Cette ouverture, en version d'abord papier puis en ligne, des RFCs a nourri l'image libertaire du réseau des réseaux et celle d'une « communauté scientifique idéale »³. Certains pères fondateurs y contribuent aussi. Elisabeth Feinler, qui rejoint en 1972 l'Augmentation Research Center pour travailler à ARPANET, sur sollicitation de Douglas Engelbart⁴, se souvient :

Le groupe de Doug [...] portait des jeans ou des salopettes, des birkenstocks, des flip-flops sans chaussettes ; ils avaient de longues barbes, des cheveux en bataille – et les femmes ne portaient probablement pas de soutien-gorge. [...] Dire que ce groupe était inhabituel pour l'époque est peu dire : ils étaient carrément bizarres !⁵.

1. SERRES, *op. cit.*, p. 447.

2. *ibid.*, p. 481.

3. FLICHY, *op. cit.*

4. Douglas Engelbart est passé à la postérité pour l'invention de la souris ou ses travaux sur les interfaces hommes-machines.

5. Elisabeth FEINLER, « The Network Information Center and its archives », dans : *IEEE Annals of the History of Computing* 32.3 (2010), p. 83–89.

Il faut toutefois se garder de faire des pionniers de l'Internet une communauté de doux rêveurs, ce sont avant tout des ingénieurs de talent. Comme le notait l'un d'entre eux, Jon Postel : « Soyez libéral dans ce que vous acceptez et conservateur dans ce que vous envoyez. »¹

2.2 Le règne du *running code* et du *rough consensus*

« Nous rejetons les rois, les présidents et les votes. Nous croyons en un consensus approximatif et un code fonctionnel »². Ces phrases célèbres de David Clark en 1992 sont fidèles à l'esprit des chercheurs qui conçurent ARPANET.

Par opposition aux « standards de droit » développés au sein des organismes de normalisation (qui restent toutefois des avis ou des recommandations qui n'ont pas de caractère contraignant), c'est par une politique de « standards de fait » que se construit Internet, bouleversant les règles du jeu de la normalisation.

Ceci ne signifie toutefois pas une absence de règles et si Netville, héritière des communautés scientifiques pionnières est en partie dans la lignée d'une tradition académique de circulation des savoirs et de « science ouverte »³, elle s'emploie aussi à définir des cadres, certes souples, mais qui passent par la structuration d'organes techniques, depuis le groupe informel de chercheurs qui forment le *Network Working Group* en 1968, en passant par la création des RFC ou la mise en place de l'INWG. Voient ensuite le jour, l'IAB (*Internet Advisory/Activities/Architecture Board*) derrière David Clark, l'IANA (*Internet Assigned Numbers Authority*) derrière Jon Postel pour le nommage et en 1992 l'ISoC (*Internet Society*) qui coordonne les efforts de groupes de réflexion technique comme l'IETF (*Internet Engineering Task Force*) ou l'IAB.

C'est dans une logique proche que se place Tim Berners-Lee pour le Web. Comme le souligne Michael L. Dertouzos, qui dirigeait alors le MIT-Laboratory for Computer Science, et a soutenu le projet : « Alors que les

1. RFC 1122.

2. Nous traduisons.

3. J. Leslie KING, E. Rebecca GRINTER et M. Jeanne PICKERING, « The rise and fall of Netville : The Saga of Cyberspace Construction Boomtown in the Great Divide », dans : *Culture of the Internet*, sous la dir. de S. KIELSER, Mahwah, NJ : Erlbaum, 1997, p. 3–33.

technologues et les entrepreneurs lançaient ou fusionnaient des sociétés pour exploiter le Web, ils semblaient fixés sur une seule question : « Comment puis-je rendre le Web mien ? ». Pendant ce temps, Tim demandait, « Comment puis-je faire le Web vôtre ? »¹

Ainsi, alors qu'au printemps 1993 l'Université du Minnesota décide que les entreprises devront payer pour utiliser Gopher² (son usage restant gratuit pour les institutions dédiées à l'éducation ou à but non lucratif), le 30 avril de la même année, à la demande de Tim Berners-Lee, le Cern³ accepte de permettre l'usage libre et gratuit des protocoles du Web. Ce choix est décisif pour l'avenir du Web mais aussi pour la structuration du W3C.

Tim Berners-Lee se tourne dès 1992 vers l'IETF. Si les activités notamment langage ne sont pas une vocation première de l'IETF (mais concernent par contre le Web avec HTML), c'est aussi la volonté de Tim Berners-Lee qui explique la décision de créer un consortium dédié au Web⁴. La première conférence internationale du *World Wide Web* se tient au Cern en mai 1994. La même année le MIT-LCS devient le premier hôte du World Wide Web Consortium (W3C).

Organisation internationale à but non lucratif, fondée pour garantir la bonne évolution du Web, inventer et promouvoir des langages et des protocoles universels, ses principes d'organisation reposent sur des choix fondamentaux comme la neutralité vis-à-vis du marché⁵ : les protocoles ne doivent pas privilégier une entreprise, tenir compte du marché, mais être adoptés selon une logique qui les rapproche d'un « bien commun ».

Comme l'a montré Andrew Russell, le W3C va choisir un habile équilibre entre centralisation et décentralisation, un modèle intermédiaire

1. Tim BERNERS-LEE, *Weaving the Web*, New York, NY : Harper Business, 2000, p. viii.

2. Avant le Web, Gopher permettait de faire des recherches de fichiers dans l'Internet. Gopher ne contenait que du texte.

3. Organisation européenne de recherche nucléaire au sein de laquelle Tim Berners-Lee a développé le Web.

4. « Je voulais un consortium qui fonctionne selon un procédé ouvert comme l'IETF, mais qui soit plus rapide et efficace, parce que nous aurions à bouger vite ». Nous traduisons. BERNERS-LEE, *op. cit.*, p. 92.

5. Cette volonté de neutralité à l'égard du marché ne doit pas être confondue avec la neutralité de l'Internet telle que nous l'entendons aujourd'hui (voir dans la suite du chapitre). Toutefois cette idée de neutralité par rapport au marché est importante encore aujourd'hui, notamment pour ceux qui craignent que l'érosion des principes de l'Internet et de la neutralité n'entraîne un ticket d'entrée pour les développeurs de services sur le Web.

entre celui de l'ouverture extrême choisi par l'IETF et celui plus fermé de l'ICANN (Internet Corporation for Assigned Names and Numbers), organisation de nommage critiquée pour son insuffisante prise en compte des différentes parties prenantes :

Le modèle du W3C est à mi-chemin entre l'IETF et l'ICANN : il s'oppose à la vitesse lente de développement par le développement de code au sein du W3C ; en incluant des membres de l'industrie ses recommandations sont davantage susceptibles d'être mises en œuvre rapidement et efficacement ; il considère aussi sérieusement et répond aux apports de ses membres et du grand public avant d'émettre ses recommandations.¹

2.3 Les limites du « gouvernement par la technique »

En se retrouvant au cœur des débats au Sommet Mondial sur la Société de l'Information (SMSI) en 2003 et 2005, le « réseau des réseaux » a vu sa dimension politique et sociale largement reconnue : en 2003, la moitié des 11 000 participants sont des représentants officiels de 175 pays mais il y a également 1 000 représentants des différents organes de l'ONU, 200 d'autres organisations, 481 ONG représentées, 1 000 personnes issues du monde des médias. Bien que le jeu soit largement biaisé par le droit de parole, l'action des acteurs non gouvernementaux permet d'intégrer davantage les enjeux sociaux dans les réflexions. Le SMSI aboutit à la création du Forum sur la Gouvernance de l'Internet et la société civile gagne une reconnaissance de son rôle dans la gouvernance de l'Internet au côté des États, des acteurs techniques et industriels.

Au SMSI, les critiques contre le leadership américain dans l'Internet, nées à la fin des années 1990, s'amplifient. Elles visent particulièrement l'ICANN, association privée à but non lucratif, dépendant du droit commercial de la Californie et liée par un protocole d'accord au Département du Commerce des États-Unis².

Créée en 1998, l'ICANN apparaît comme le symbole d'une gouvernance trop unilatérale de ressources, qui constituent pourtant un enjeu

1. Andrew RUSSELL et Laura DENARDIS, « The W3C and Its Patent Policy Controversy : A Case Study of Authority and Legitimacy in Internet Governance », dans : *Opening Standards : The Global Politics of Interoperability*, Cambridge, MA : MIT Press, 2011.

2. Cet accord (*Memorandum of Understanding*) a pris fin en 2009.

majeur : elle est chargée d'administrer le système de nommage et d'adressage, le DNS (*Domain Name System*), qui permet d'établir la correspondance entre une adresse IP et un nom de domaine. Les critiques visent le poids insuffisant accordé aux gouvernements et aux utilisateurs dans les décisions, malgré l'existence de comités représentatifs¹, la forte domination des États-Unis, la faible représentativité des participants, l'insuffisante indépendance des registres régionaux qui voient progressivement le jour, ou encore le « monopole » de VeriSign, société américaine, à laquelle l'ICANN a confié la gestion du .net, du .com et de la racine (ce qui lui donne pouvoir sur l'annuaire de l'Internet et lui assure une manne financière). Comme le note Françoise Massit-Folléa :

L'association californienne de droit privé ICANN, dont les décisions ont vocation à s'imposer à la planète entière, est en quelque sorte un *objet juridique non identifié* :

- Ce n'est pas une instance de normalisation technique : celle-ci se fait en amont et en aval de la mission de l'ICANN, via des associations informelles (par ex. l'IETF) ou plus formelles (par ex. le W3C) ;
- Ce n'est pas une agence gouvernementale : les accords successifs avec le DoC (succession de Memorandum of Understanding et actuel Affirmation of Commitments) sont toujours présentés comme un progrès vers la *privatisation* de l'organisme ;
- Ce n'est pas non plus une agence intergouvernementale : le comité des gouvernements (imposé initialement par l'Europe) n'y a qu'un rôle consultatif ; il n'existe pas de lien institutionnel avec l'ONU et l'ICANN est en rivalité permanente avec l'Union Internationale des Télécommunications. Pourtant l'ICANN a contracté directement avec l'OMPI² un mode arbitral de résolution des conflits sur les noms de domaine, l'UDRP³ ;
- Ce n'est pas une banale association à but non-lucratif : l'objectif est de gérer une *ressource d'intérêt général* mais le marché des noms de domaine est une activité fort rémunératrice. Or

1. Ce sont respectivement le GAC, Governmental Advisory Committee, et At-Large.

2. Office Mondial de la Propriété Intellectuelle.

3. *Uniform Domain Name Dispute Resolution Policy*. La question du droit des marques au niveau international s'est déjà posée pour la radio et la télévision et il paraît surprenant que l'Internet ait suscité une démarche spécifique. Cf. l'interview de Michaël Krieger réalisée en 2000 pour la revue *Cyber.Law* de Harvard. <http://cyber.law.harvard.edu/is99/governance/krieger.html>.

l'un des points faibles de l'ICANN a toujours été son manque d'*accountability* ;

- Ce n'est pas, enfin, une organisation « ouverte » : parmi la pléthore de ses composantes, celle qui est censée représenter les usagers pèse bien peu sur les choix stratégiques ; d'ailleurs après l'échec, en octobre 2000, de la *première élection électronique mondiale* qui devait en désigner les membres, les leçons n'ont guère été tirées en interne.¹

Malgré la participation de multiples entités nationales, régionales, internationales à la gouvernance, les Européens, qui ont eu une position parfois jugée trop suiviste, appellent aujourd'hui à une reconfiguration de la répartition des pouvoirs. Ils ne sont pas seuls dans cette lutte depuis les revendications du Groupe des 77 et de la Chine² jusqu'aux interventions actuelles des BRIC (Brésil, Russie, Inde, Chine et Afrique du Sud) dans les débats.

L'UIT, agence de l'ONU, a mal accepté d'avoir vu la régulation technique de l'Internet lui échapper. Ses relations avec l'ICANN passent donc par des phases de tensions plus ou moins vives. Mais elle représente encore l'horizon *régulateur* de ceux qui, au nom des prérogatives des États souverains, envisagent l'internationalisation de la gestion technique de l'internet dans le cadre formel d'une instance multilatérale.³

Dans un article du *New York Times* du 24 mai 2012, *Keep the Internet Open*, Vinton Cerf propose un plaidoyer en faveur de l'ouverture, non dénué d'arrière-pensées : en particulier, il critique la volonté de l'UIT de s'investir davantage dans la gouvernance de l'Internet, affirmant que l'ouverture est du côté des organes nés en son sein plutôt que de l'institution internationale.

À l'heure actuelle, l'UIT met l'accent sur les réseaux de télécommunication et sur l'attribution des fréquences radio plutôt que sur

1. Françoise MASSIT-FOLLÉA, « La gouvernance de l'Internet. Une internationalisation inachevée », dans : *Le Temps des médias* 18.1 (2012), p. 29–40, URL : <http://www.cairn.info/resume.php?IDARTICLE=TDM0180029>, p. 34.

2. Le groupe des 77 est un regroupement de pays en développement formé en 1964 pour porter les intérêts de ses membres à l'ONU.

3. MASSIT-FOLLÉA, *op. cit.*, p. 36.

l'Internet *per se*. Certains membres visent à élargir le champ d'action de l'Union pour y inclure la régulation d'Internet. Chacun des 193 membres obtient un vote, peu importe son bilan en matière de droits fondamentaux — et une majorité simple suffit à provoquer un changement. Les négociations ont lieu principalement entre les gouvernements, avec un accès très limité à la société civile ou d'autres observateurs. [...] En revanche, l'UIT crée des obstacles importants à la participation de la société civile.¹

Vinton Cerf renvoie vers l'UIT les critiques qui sont parfois adressées aux instances de gouvernance de l'Internet et réaffirme le « *We reject Kings, presidents and voting* » de David Clark... une idéologie libérale... qui peut contribuer à faire le jeu du libéralisme.

2.4 *Where does 800 pound-gorillas sit ? Anywhere they want !*

Le principe de décentralisation gouverne aux origines de l'Internet, à un moment où ses principaux usagers sont les chercheurs. Cependant, le modèle va être érodé par le passage au grand public et à l'Internet commercial dans les années 1990, au profit d'architectures du type client-serveur. Depuis le navigateur Web, beaucoup d'applications se basent sur un système où le client amorce une connexion à un serveur connu, en télécharge des données et se déconnecte. Des services comme les réseaux sociaux, les outils de messagerie, les applications de stockage de contenus se fondent sur des modèles techniques et économiques dans lesquels les utilisateurs demandent des services à de puissants serveurs qui stockent de l'information et/ou gèrent le trafic. Internet devient de plus en plus asymétrique. Le déploiement de pare-feu, l'augmentation des adresses IP dynamiques et la popularité de la *Network Address Translation* érodent également la symétrie du réseau tandis que le *best effort* recule, afin de résoudre des problèmes de congestion, de créer des priorités ou de ralentir certains paquets de données (vidéo, VoIP...). Peut-on ralentir des trafics fortement consommateurs de bandes passantes, liés à des usages pair-à-pair ou de téléchargement de contenus lourds ? Est-il opportun ou possible de favoriser des données, par exemple celles pour lesquelles un éditeur de contenus aurait payé des marges arrières à un fournisseur d'accès à Internet ? Ces

1. <http://www.nytimes.com/2012/05/25/opinion/keep-the-internet-open.html>.

questions renvoient à la Neutralité de l'Internet que Lawrence Lessig et Tim Wu portaient dès 2002 sur la scène politique, publique et médiatique.

Modifier le transport des données sur l'Internet, c'est aussi modifier l'accès aux services et aux applications. Tim Berners-Lee dans un article de *Scientific American* du 4 mai 2011, *Long Live the Web : A Call for Continued Open Standards and Neutrality*, souligne à quel point leur avenir est lié, car tout usager du Web est un usager d'Internet¹. Aussi, une réflexion sur où commence et s'arrête la question de la neutralité (faut-il considérer l'ensemble de la chaîne de valeur ou le simple acheminement des données ?) est essentielle et certains suggèrent de privilégier l'expression d'« Internet ouvert » plutôt que celle d'« Internet neutre ».

L'Internet ouvert renvoie à un espace qui n'est sous le contrôle d'aucun acteur unique, où chacun peut librement créer, entreprendre et voir son expression, ses créations, son activité accessibles à l'ensemble de l'Internet²

Cette notion est investie de valeurs plus globales (liberté d'entreprise, d'expression...)

Mais David Clark souligne toutefois à quel point les avocats de l'ouverture, n'en défendent pas nécessairement les valeurs « sociales », à l'instar des grands acteurs industriels³. Ceux qu'il appelle les *800-pound gorillas*, en référence à une expression américaine « Where do 800-pound gorillas sit ? Anywhere they want ? »⁴, en promouvant l'ouverture, pensent surtout à leurs revenus et au futur de leur industrie.

Conclusion

L'ouverture est un concept polyvalent, désignant tout à la fois les spécifications techniques et les modes de gouvernance. Cette polysémie est présente dès l'origine d'ARPANET et bien qu'aujourd'hui beaucoup d'ac-

1. <http://www.scientificamerican.com/article.cfm?id=long-live-the-web>.

2. Consultation publique sur la « Neutralité du Net », Synthèse des réponses, 2010. Voir <http://www.mediapart.fr/files/synthese-consultation-neutralitenet.pdf>.

3. CLARK, *op. cit.*

4. « Où s'asseient les gorilles de 800 livres ? Où ils veulent » est une expression imagée pour montrer la toute puissance des multinationales.

teurs se réclament de l'ouverture, comme le souligne David Clark, celle-ci se charge de nuances complexes qui peuvent conduire à des résultats très différents. Pour preuve la définition on ne peut plus nuancée de l'ouverture par Neelie Kroes, commissaire européenne chargée de la société numérique, à la conférence du W3C d'avril 2012 :

Après avoir rappelé son attachement à l'ouverture d'Internet, en déclarant dès la première phrase du discours qu'il s'agit *de la meilleure chose* en matière de réseau, Neelie Kroes vient rapidement modérer son propos : « L'ouverture est aussi complexe car sa signification est parfois peu claire ». [...] Ça se corse davantage avec la fourniture de l'accès à Internet. Pour Neelie Kroes, la notion d'ouverture est ici particulièrement « subtile ». [...] Elle va ici plus loin, se disant « engagée à garantir la neutralité du Net » d'un côté et refusant de l'autre à assimiler ce combat au « bannissement de toutes les offres ciblées ou limitées ». L'ouverture consistant ici pour elle davantage à « être transparent » sur ces offres, et à « laisser aux consommateurs la possibilité de choisir librement et aisément s'ils les souhaitent ou non. »¹

De là à penser que l'ouverture est fragile et ses motivations hétéroclites, parfois loin des aspects éthiques et idéologiques qu'on veut lui attribuer, il n'y a qu'un pas... franchi par Linus Torvalds récemment :

L'*open source* ne fonctionne vraiment que si chacun contribue pour ses propres raisons égoïstes, et celles-ci ne sont pas forcément de l'ordre d'une « récompense financière », a insisté M. Torvalds. Il a lui-même commencé à créer Linux pour « le plaisir de bricoler ». Programmer était sa passion. Apprendre à contrôler le matériel était son but égoïste. [...] Et pour mener à bien un tel projet *open source*, « la confiance est essentielle ». Personnellement, a expliqué le programmeur, « je veux que les gens aient confiance dans le fait que je suis impartial pas seulement parce qu'ils m'ont vu maintenir le noyau (*kernel*) au fil des ans, mais parce qu'ils savent que je n'ai tout simplement pas d'incitations qui pourraient m'entraîner à soutenir telle entreprise spécialisée Linux plutôt qu'une autre. »²

1. <http://owni.fr/2012/04/19/leurope-delaissela-neutralite-du-net>.

2. <http://www.silicon.fr/linus-torvalds-linux-75664.html>.

Les réactions aux propos de Linus Torvald ont été suffisamment violentes et irritées pour témoigner de la charge idéologique dont est aujourd'hui investie l'ouverture. Celle-ci fut toutefois à l'origine aussi très pragmatique dans les réseaux, comme le montre l'exemple de Cyclades. Reste à voir si elle pourra, dans le domaine des réseaux, résister à la volonté d'introduire de plus en plus d'éléments qui étioient l'architecture originelle de l'Internet (des mesures en faveur de la Qualité de Service jusqu'aux écosystèmes fermés – Facebook, Apple), et à l'indifférence d'utilisateurs qui, pour beaucoup, se satisfont et même recherchent des systèmes fermés, ces jardins privés et silos d'enfermement que contribuent à créer de plus en plus les empires du Net.

Ceci pourrait ouvrir une nouvelle page de l'histoire de l'Internet, à moins que l'on ne préfère réécrire déjà son passé, à l'instar de Gordon Crovitz, qui, le 22 juillet 2012 dans le *Wall Street Journal*, attribuait finalement la réussite de l'Internet à Xerox.

Si le gouvernement n'a pas inventé l'Internet, qui est-ce ? Vinton Cerf a élaboré le protocole TCP/IP, l'épine dorsale de l'Internet, et Tim Berners-Lee obtient le crédit pour les hyperliens. [...] Mais le crédit va pleinement à la société où M. Taylor a travaillé après avoir quitté l'ARPA : Xerox. C'est dans les laboratoires Xerox PARC dans la Silicon Valley dans les années 1970 que l'Ethernet a été développé pour relier différents réseaux informatiques. Les chercheurs y ont également développé le premier ordinateur personnel (le Xerox Alto) et l'interface utilisateur graphique qui anime encore l'utilisation des ordinateurs aujourd'hui. [...] Il est important de comprendre l'histoire de l'Internet parce qu'elle est trop souvent à tort invoquée pour justifier un gouvernement fort. Il est également important de reconnaître que la construction de grandes entreprises technologiques exige à la fois l'innovation et les compétences nécessaires pour apporter ces innovations sur le marché. Comme le montre le contraste entre Xerox et Apple, peu de chefs d'entreprise réussissent dans ce défi. Ceux qui le font — et non le gouvernement — méritent le crédit pour ce qu'ils réalisent.

Une réécriture de l'histoire tout entière dédiée à la gloire du privé, où l'on retrouve Xerox, mais plus vraiment sous l'angle stallmanien de la fermeture... L'histoire, fille ou victime de son temps ?

Références

- ABBATE, Janet, *Inventing the Internet*, Cambridge, MA : MIT Press, 1999.
- BELTRAN, Alain et Pascal GRISET, *Histoire d'un pionnier de l'informatique, 40 ans de recherches à l'Inria*, Les Ulis : EDP France, 2007.
- BERNERS-LEE, Tim, *Weaving the Web*, New York, NY : Harper Business, 2000.
- BYGRAVE, Lee et John BING, *Internet Governance. Infrastructure and Institutions*, Oxford : Oxford University Press, 2009.
- CERUZZI, Paul E., « Aux origines américaines de l'Internet : projets militaires, intérêts commerciaux, désirs de communauté », dans : *Le Temps des Médias* 18.1 (2012), p. 15, URL : <http://www.cairn.info/resume.php?IDARTICLE=TDM0180015>.
- CLARK, David, « Network Neutrality : Words of Power and 800-Pound Gorilla », dans : *International Journal of Communication* 1 (2007), p. 701–708, URL : <http://groups.csail.mit.edu/ana/Publications>.
- DENARDIS, Laura, *Opening Standards : The Global Politics of Interoperability*, Cambridge Mass. : MIT Press, 2011.
- FEINLER, Elisabeth, « The Network Information Center and its archives », dans : *IEEE Annals of the History of Computing* 32.3 (2010), p. 83–89.
- FLICHY, Patrice, « Internet ou la communauté scientifique idéale », dans : *Réseaux* 17.97 (1999), p. 77–120.
- *L'imaginaire d'Internet*, Paris : La Découverte, 2001.
- GOLDSMITH, Jack et Tim WU, *Who controls the Internet ? Illusions of a Borderless World*, Oxford : Oxford University Press, 2006.
- KELTY, Christopher, *Two bits : the cultural significance of free software*, Durham : Duke University Press, 2008.
- KING, J. Leslie, E. Rebecca GRINTER et M. Jeanne PICKERING, « The rise and fall of Netville : The Saga of Cyberspace Construction Boomtown in the Great Divide », dans : *Culture of the Internet*, sous la dir. de S. KIELSER, Mahwah, NJ : Erlbaum, 1997, p. 3–33.
- LE CROSNIER, Hervé et Valérie SCHAFER, *La neutralité de l'Internet, un enjeu de communication*, Paris : CNRS Edition, 2011.
- MASSIT-FOLLÉA, Françoise, « La gouvernance de l'Internet. Une internationalisation inachevée », dans : *Le Temps des médias* 18.1 (2012),

- p. 29–40, URL : <http://www.cairn.info/resume.php?IDARTICLE=TDM0180029>.
- MUSIANI, Francesca et Valérie SCHAFER, « Le modèle Internet en question (années 1970-2012) », dans : *Flux* 3.85 (2011), p. 62–71.
- POUZIN, Louis, « A Proposal for Interconnecting Packet Switching Networks », dans : *Proceedings of EUROCOMP*, Brunel University, 1974, p. 1023–1036.
- QUATERMAN, John, *The Matrix : Computer Networks and Conferencing Systems Worldwide*, Digital Press, 1989.
- RUSSELL, Andrew, *An Open World : History, Ideology, and Network Standards*, Cambridge, MA : Cambridge University Press, à paraître.
- RUSSELL, Andrew et Laura DENARDIS, « The W3C and Its Patent Policy Controversy : A Case Study of Authority and Legitimacy in Internet Governance », dans : *Opening Standards : The Global Politics of Interoperability*, Cambridge, MA : MIT Press, 2011.
- SALTZER, John, David REED et David CLARK, « End-to-End Arguments in System Design », dans : *Second International Conference on Distributed Computing Systems*, Versailles, 1981, p. 509–512, URL : <http://web.mit.edu/Saltzer/www/publications/endtoend/endtoend.pdf>.
- SALUS, Peter, *A Quarter Century of Unix*, Reading : Addison-Wesley, 1994.
- SCHAFER, Valérie, *La France en réseaux*, Paris : Editions Nuvis, 2012.
- SERRES, Alexandre, « Aux sources d'internet : l'émergence d'ARPA-NET », Thèse en Sciences de l'information et de la communication, Rennes : Université de Rennes, 2000.
- VAN SCHEWICK, Barbara, *Internet Architecture and Innovation*, Cambridge, MA : MIT Press, 2010.
- WILLIAMS, Sam, Richard STALLMAN et Christophe MASUTTI, *Richard Stallman et la révolution du logiciel libre. Une biographie autorisée*, Paris : Eyrolles, 2010.

Ingénieurs, hackers : naissance d'une culture

Christophe MASUTTI

Au début des années 1960, lorsque l'IBM/360 fut prêt à être produit et vendu à plusieurs centaines d'exemplaires, il était fort présomptueux d'affirmer que le marché était lui aussi prêt à accueillir ces nouvelles machines qui tiraient parti de plusieurs innovations matérielles assez peu connues. Néanmoins, ces machines étaient présentes et leur création relevait davantage d'une démonstration des capacités d'innovation des ingénieurs d'IBM que de l'analyse d'un besoin concrètement exprimé dans les secteurs industriels ou de services. Nous verrons dans cet article à quel point l'organisation de l'activité d'ingénierie a modelé l'industrie informatique. Cela tient au fait que l'informatique des décennies 1950 et 1960 ne fut pas seulement un « âge d'or » pour la recherche et le développement, mais un moment où se cristallisèrent socialement un ensemble de pratiques et de représentations à priori hétérogènes. Ces dernières déterminèrent nombre d'innovations concernant le partage de l'information, l'ingénierie logicielle, les modèles d'organisation des entreprises, l'utilisation des machines et leurs trajectoires.

Dans ce contexte, l'adage suivant lequel « la nécessité crée l'innovation » est quelque peu désuet. Dans bien des cas, dans l'histoire de l'informatique moderne et particulièrement dans l'évolution des programmes et des langages de programmation, la « nécessité » n'est que peu ou pas

exprimée socialement. Elle le fut par exemple dans le cas de la cybernétique pour la création du langage LISP, aujourd'hui l'un des plus anciens langages de haut niveau inventé par John McCarthy en 1958. Ce langage répondait à un besoin : dépasser la limite des capacités humaines pour se représenter des programmes de plus en plus complexes. Or, l'évolution de ce langage et les multiples branches qui en découlèrent, y compris l'invention des Machines LISP spécialement conçues, furent davantage le produit des efforts de « rivalités » entre hackers motivés par « l'émulation procurée par l'habileté technique, caractéristique de la culture hacker », pour reprendre les termes de G. Steel et R. Gabriel analysant l'évolution de LISP depuis sa création jusqu'aux années 1990¹.

Cet article vise à explorer les fondements de la culture hacker et à expliquer comment, à partir d'un modèle d'organisation, des principes d'action ont pu voir le jour, se réclamant de l'éthique hacker et aujourd'hui exprimés plus prosaïquement par le « mouvement du Libre ».

Nous resterons attachés à la période des deux décennies 1950 et 1960 où nous verrons d'une part comment le concept de *timesharing* a « révolutionné » les représentations de l'ingénierie, en particulier dans le secteur industriel. D'autre part, nous tâcherons de déterminer quels furent les modèles d'organisation, hérités de l'industrie, qui entrèrent en jeu lors de l'émergence de l'ingénierie de programmation (*software engineering*). Cette analyse nous conduira à nous interroger sur le hacker en tant qu'archétype de l'ingénieur-programmeur, dont les pratiques (ou l'*ethos*) qu'il véhicule impliquent en permanence un positionnement par rapport aux enjeux du changement technologique dont il est à la fois concepteur et utilisateur.

1. À propos d'ordinateurs et de temps partagé

En février et mars 1957, un ingénieur issu de l'aéronavale, Robert (Bob) Bemer, publie un article en deux parties dans la revue *Automatic Control*,

1. Guy L. STEELE JR. et Richard P. GABRIEL, « The Evolution of Lisp », dans : *The second ACM SIGPLAN conference on History of programming languages*, HOPL-II, New York, NY, USA : ACM, 1993, 231–270, URL : <http://doi.acm.org/10.1145/154766.155373>.

qu'il intitule « Qu'est-ce qu'un ingénieur doit savoir à propos de la programmation informatique ? »¹. Bemer est un acteur célèbre dans le monde informatique. Il travailla notamment avec Grace Hopper sur le langage COBOL et a contribué significativement aux spécifications de l'ASCII, ce qui lui vaut d'être parfois surnommé le « père de l'ASCII ». Dans cet article, Bemer se penche sur le changement des pratiques qu'implique l'apparition de l'informatique dans le monde de l'ingénierie. Pour lui, un ingénieur ne doit pas seulement concevoir des modèles et attendre qu'un programmeur professionnel crée les programmes qui permettent de faire de multiples calculs et les entre dans la machine. L'ingénieur doit au contraire connaître la programmation car, de cette manière, il peut créer et améliorer des langages de programmation adaptés aux besoins exacts de sa spécialité d'ingénierie. Cela, il doit savoir le faire sur des machines (à temps) partagées(é), c'est-à-dire ces gros ordinateurs *mainframe* très chers dont la particularité est de permettre à plusieurs utilisateurs d'avoir accès au temps de calcul.

Le premier avantage de cette mutualisation de la machine est de partager le coût du temps de calcul, compte tenu de l'investissement financier qu'elle représente. Le second avantage est que, compte tenu des innovations de ces machines, la création de nouveaux langages de programmation permet de rendre de plus en plus efficaces leurs applications à l'ingénierie. La programmation étant elle-même une forme d'ingénierie, les apprentissages de ces langages se font de manière coopérative, entre les utilisateurs. De ce partage des langages (qui s'exprime surtout par le partage de compilateurs sans cesse améliorés), on apprend donc que l'organisation de la communauté et la structure des machines (ici, les ordinateurs à temps partagé), favorisent l'émergence des formes de coopérations qui transcendent les différents domaines de l'ingénierie où l'informatique peut s'appliquer : Bemer cite alors 80 secteurs classés dans 7 domaines (aéronautique, chimie, mathématiques, électricité, physique (nucléaire), statistique, divers) où la programmation est censée devenir un savoir-faire incontournable.

1. Robert W. BEMER, « What the Engineer Should Know Computer Programming », dans : *Automatic Control Magazine* 2, Feb./Mar. (1957) (document aimablement fourni par le Charles Babbage Institute).

En réalité, Bemer se positionnait en visionnaire. En 1957, les ordinateurs à temps partagés qui pouvaient être ainsi utilisés n'existaient encore que dans l'esprit de certains programmeurs de génie, parmi lesquels Bob Bemer et John McCarthy, l'inventeur du langage LISP et, justement, du premier système d'exploitation à temps partagé. Il est certain qu'une page décisive était en train de se tourner. Bemer avait exprimé les besoins et la manière d'y répondre. À peine deux ans plus tard, une rencontre entre industriels et chercheurs du MIT, à l'occasion d'une démonstration de LISP¹, donna le jour au concept de *computer utility* : la possibilité de livrer des services en séparant les ressources matérielles et le calcul. Sur la base de cet argument, des fonds soulevés auprès de la National Science Foundation permirent à l'équipe de McCarthy de créer en 1961 le premier système d'exploitation (ou environnement) pour le temps partagé : Compatible Time-Sharing System (CTSS). C'est sur la base de cette nouveauté pour l'esprit que représentait un langage informatique autant qu'à travers la vision d'un monde de machines au service de l'innovation technologique que naquirent plus tard les premières sociétés de services fournissant aux institutions comme aux particuliers des accès sur des machines distantes pouvant supporter plusieurs utilisateurs simultanés.

Les épisodes que nous venons de raconter illustrent combien l'histoire de l'informatique moderne ne saurait se contenter de retracer les grandes étapes des créations de machines et de langages, d'un point de vue finaliste, comme pour arriver enfin à la définition contemporaine d'un ordinateur. Selon l'historien Paul E. Ceruzzi, s'intéresser à l'histoire des ordinateurs consiste d'abord à admettre que la définition même d'un ordinateur implique une période durant laquelle il est possible de faire une histoire des techniques informatiques. Un ordinateur sert à calculer. De ce point de vue, l'histoire des outils techniques servant au calcul peut nous emmener loin dans le temps, au moins pour couvrir la période séparant les premiers bouliers-compteurs (manuel) à la Pascaline (automatisme). Il en va différemment pour les ordinateurs, que Ceruzzi nomme *modern computer* et définit ainsi :

Il s'agit d'un système : un dispositif hiérarchique composé de matériel informatique (hardware) et de logiciels (software). Quiconque

1. John MCCARTHY, « History of LISP », dans : *ACM-SIGPLAN Notices* 13 (1978), 217–223.

travaille sur ce système à un certain niveau n'a pas de visibilité sur ce qui se passe dans les autres niveaux. Quant aux plus hauts de ces derniers, ils sont constitués de logiciels – par définition, les éléments qui n'ont pas de forme tangible mais que l'on peut au mieux décrire comme étant les méthodes d'organisation.¹

La complexité des ordinateurs, leurs représentations, leurs agencements et les connaissances nécessaires à leur élaboration et leur utilisation, toutes ces facettes justifient qu'une histoire cohérente des ordinateurs modernes ne peut se réduire à l'addition de plusieurs histoires comme celle de l'électronique, de l'ingénierie, du calcul informatique, ou de l'électricité, prenant en compte l'objet « ordinateur » comme une innovation parmi d'autres. En somme, l'histoire des ordinateurs ne saurait se contenter du seul point de vue de l'histoire informatique analysant une production issue de l'ingénierie et de la technique. Dans ce dernier domaine, en effet, il serait difficile de réduire les facteurs décisifs de l'innovation à la seule dynamique de l'offre et de la demande.

L'analyse politique permet de rapprocher les innovations en techniques de l'information avec la demande, par exemple, en matière de politique de défense et d'armement. En revanche, toujours en suivant cet exemple, comment faire le lien entre les politiques publiques encourageant massivement les centres de recherches et l'accroissement de nouvelles machines, notamment durant la période de la Guerre froide ? La compétition internationale en matière de puissance de calcul, dont le Plan Calcul lancé en 1966 en France par le Général De Gaulle, démontre la recherche d'indépendance par rapport aux autres pays plus novateurs (après le rachat de Bull par General Electric), mais justifie-t-il à lui seul le développement industriel français dans le domaine des circuits intégrés ? Là encore, ce serait faire un raccourci que de ne considérer les résultats de la recherche et de l'ingénierie uniquement comme les fruits de la volonté politique. Il existe en effet une foule de facteurs sociaux déterminant l'innovation dont souvent la volonté politique ne fait que formaliser rétrospectivement les moyens humains et cognitifs mobilisés.

Il en va ainsi du développement des ordinateurs à temps partagé. Le fait d'avoir besoin de pouvoir accéder à plusieurs, via des terminaux d'ac-

1. Paul E. CERUZZI, *A History of Modern Computing*, Cambridge, Mass. : MIT Press, 1998, p. 4.

cès, aux ressources d'une seule machine peut être interprété de plusieurs manières :

- La réponse à un besoin d'une population grandissante de chercheurs contraints à effectuer des calculs toujours plus complexes sur des machines coûteuses et ne pouvant pas assurer la charge de plus d'un utilisateur à la fois. Il s'agirait donc d'optimiser l'accès aux ressources.
- Une solution envisagée de manière à pouvoir optimiser le développement de nouveaux langages informatiques de haut niveau, comme le langage LISP, mais aussi tous les langages servant à modéliser dans différents secteurs d'ingénierie.
- Enfin, il est possible aussi de voir dans l'émergence des ordinateurs à temps partagé le résultat de plusieurs années de recherche en matière d'ingénierie logicielle visant à créer des systèmes d'exploitation capables de gérer efficacement les accès (comme ce fut le cas du projet MAC, qui donna le système Multics, successeur de CTSS, et qui deviendra encore plus tard Unix). Une linéarité est alors envisagée entre la recherche, les applications et l'industrie, constituant généralement l'interprétation la plus facile et la plus connue de l'histoire des ordinateurs à temps partagé.

Peut-on établir une hiérarchie stricte entre ces facteurs ? Aucunement. Le fait est que les déterminants sociaux de telles innovations sont complètement imbriqués : les chercheurs ne font pas que chercher, il faut qu'ils financent leurs projets de recherche et produisent l'expertise nécessaire à la décision publique en répondant à des appels d'offres ; les spécialistes de l'ingénierie logicielle, les programmeurs, ne font pas que s'adapter au matériel existant, ils produisent eux-mêmes des savoirs-faire et créent une demande matérielle, quant aux politiques publiques et aux institutions, elles formalisent dans une logique de projets et de financement les orientations pré-existantes ou, sur la base de l'expertise des chercheurs et des ingénieurs, accentuent des orientations méritant des développements plus poussés.

Le terme « révolution » employé dans plusieurs domaines de l'informatique moderne est souvent utilisé pour expliquer une convergence entre un besoin et l'émergence d'une réponse technique qui s'impose dans de multiples voire tous les domaines d'activités. Il s'agit d'une construction sociale, pour reprendre l'expression consacrée. Mais cette image est quelque

peu faussée. Par exemple, l'anthropologue Bryan Pfaffenberger¹ montre que ce sont les industriels qui ont construit le mythe de l'utilité révolutionnaire de l'ordinateur personnel afin d'en lancer le développement et la vente à grande échelle et définir les contours du marché.

La « révolution » informatique est donc beaucoup plus théorique. Elle s'est exprimée à partir du moment où naquirent de nouveaux apprentissages issus de l'application de l'informatique aux activités humaines et, donc, un nouveau champ épistémologique à explorer sur la base d'un rapport nouveau entre calcul et information. Loin d'être uniquement l'émergence de nouvelles technologies censées optimiser les activités existantes, ce changement s'est aussi exprimé d'un point de vue social, car il fut le vecteur de nouveaux rapports organisationnels. À propos des « acteurs » (sociaux) de la révolution informatique, Pfaffenberger – nous le suivons sur ce point dans le domaine de la sociologie des sciences et des techniques – préfère de loin utiliser non pas le terme d'acteur-réseau que l'on trouve chez Michel Callon et Bruno Latour, mais celui mieux choisi d'ingénieur hétérogène (défini par John Law), c'est-à-dire un acteur qui ne crée pas seulement une nouvelle technologie, mais qui définit un nouveau cadre du rôle social, des représentations et des valeurs de cette technologie et des ingénieurs qui l'emploient et la diffusent. C'est exactement ce que Janet Abbate propose dans un plaidoyer pour une étude STS (Science and Technology Studies) de l'histoire d'Internet : « Lorsqu'on observe à la fois les créateurs et les utilisateurs de matériels et de logiciels, on s'aperçoit que l'infrastructure et la culture ont amorcé une fusion »². Or, dans l'histoire de la révolution informatique, cette fusion est triple car elle engage à la fois l'innovation technologique, le cadre culturel et la figure de l'ingénieur hétérogène qui incarne cette culture et qui fut autrement nommé hacker, c'est-à-dire un programmeur dont le génie ne réside pas seulement dans la qualité de son travail mais aussi dans sa manière de traduire un ensemble de valeurs de la sphère informatique à la sphère sociale : le partage de l'information, l'esprit d'innovation (d'amélioration),

1. Bryan PFAFFENBERGER, « The Social Meaning of the Personal Computer : or, Why the Personal Computer Revolution Was not a Revolution », dans : *Anthropological Quarterly* 61.1 (1988), p. 39–47.

2. Janet ABBATE, « L'histoire de l'Internet au prisme des STS », dans : *Le Temps des médias* 18.1 (2012), p. 170, URL : <http://www.cairn.info/revue-le-temps-des-medias-2012-1-p-170.htm>.

les libertés d'usages de machines, l'idée qu'un « progrès technique » est aussi un « progrès social » dans la mesure où tout le monde peut y contribuer et en être bénéficiaire.

Tout cela suppose un modèle d'organisation dans le travail de ces ingénieurs hétérogènes, un modèle qui n'est pas apparu ex-nihilo, mais, comme nous allons le voir, est issu du monde industriel, et même hérité des infrastructures informatiques, comme le percevait brillamment Bob Bemer.

2. Mouffettes et canard sauvages

À partir de la fin de la Seconde Guerre mondiale, on voit apparaître dans l'industrie aéronautique américaine l'expression *shunk works* (littéralement, travaux de mouffette). Il s'agit de désigner le travail d'équipes formées d'ingénieurs, souvent cooptés par leurs pairs en fonction des besoins et des compétences, travaillant sur des projets non-officiels, c'est-à-dire soit des projets secrets relevant de la Défense nationale, soit des projets hautement novateurs dont le débouché n'est pas clairement défini par l'entreprise mais disposant de suffisamment d'éléments probants pour faire l'objet d'un investissement¹. Les *shunk works* ont emprunté leur appellation officielle au sein de l'entreprise américaine Lockheed Martin (anciennement Lockheed Corporation, qui fusionna en 1994 avec Martin Marietta), aujourd'hui la principale firme de production d'armement de la Défense américaine. Si cette appellation remonte à l'immédiate après-guerre, notamment dans le domaine des avions longs courriers, elle fourbit ses armes durant la Guerre froide, dans le contexte très concurrentiel de la course à l'armement, où l'aéronautique militaire américaine se devait de produire des avions inédits tels le Lockheed U-2, un avion de reconnaissance à très grand rayon d'action, fleuron de l'innovation technologique américaine. Dans ce contexte où l'espionnage industriel était une préoccupation majeure², la Lockheed Corporation se devait de conserver secrets certains développements quitte à ne pas élaborer de stratégie

1. Ben R. RICH et Leo JANOS, *Skunk Works : A Personal Memoir of My Years of Lockheed*, 1st Pbk. Ed, Back Bay Books, fév. 1996.

2. *ibid.*, p. 6. B. Rich rapporte sur ce point : « Nous pensions que le KGB connaissait nos principaux numéros de téléphone, et des dispositifs d'enregistrement informatisés devaient probablement s'allumer à bord de ces chalutiers [au large des côtes californiennes] lorsque

claire pouvant révéler des intentions sur la création de prototypes. Les *shunk works* permettaient à la fois d'allier la créativité, en favorisant des pratiques communautaires d'échanges de connaissances et d'expériences, et la protection industrielle, en permettant des développements n'obéissant à priori à aucune stratégie explicite et ne répondant à aucun appel d'offres gouvernemental (du moins pas officiellement, à l'image du projet XP-80, premier prototype d'avion à réaction). L'expression fut néanmoins connue dans différents milieux industriels et employée jusqu'à nos jours, de manière courante, bien que Lockheed Martin ait déposé cette marque (avec son logo, une moufette) comme étant l'autre nom de son antenne de recherche et développement, l'Advanced Development Programs (ADP).

Les *shunk works* sont donc des dispositifs d'organisation en ingénierie¹ alliant quatre principes : la vitesse de conception, l'identification de l'individu au groupe dont il fait partie et avec qui il partage une vision positive des activités en cours, la reconnaissance d'un leader (non coopté mais institué en tant que tel par la firme et sur lequel repose la responsabilité des travaux), la motivation suscitée non par l'impératif de réussite de la mise en œuvre mais par le travail lui-même porteur de changement et de nouveauté. Ainsi, les *shunk works* constituent des groupes de travail avec une forte cohésion jouant le rôle de laboratoires de recherche mais dont les larges marges de manœuvre sont stratégiquement établies par la firme. Pour autant, l'autonomie n'est pas synonyme d'indépendance : l'évaluation des travaux s'établit sur le long terme au vu des apports théoriques et pratiques du groupe de travail à la firme. En cas d'insatisfaction, un tel groupe peut être dissous ou les travaux en cours peuvent être réorientés en vue d'atteindre des objectifs différents. En retour, ce groupe opérant en dehors de l'organisation « normale » de l'entreprise est enclin à se sous-

ces téléphones sonnaient. Les services secrets américains avaient régulièrement intercepté des références aux *shunk works* au travers des communications satellites soviétiques, probablement parce qu'il n'y avait pas de traduction en russe pour ce surnom si particulier. Notre nom officiel était : Lockheed's Advanced Development Projects. »

1. Shane Greenstein, spécialiste en économie de l'innovation en informatique, montre comment les *shunk works* (et les *wild ducks*, que nous verrons plus loin) sont deux modèles pertinents pour analyser les mécanismes d'innovation au sein de la DARPA dans les projets qui donnèrent naissance à Internet. Voir Shane GREENSTEIN, « Nurturing the Accumulation of Innovations : Lessons from the Internet », dans : *Accelerating Energy Innovation : Insights from Multiple Sectors*, Univ. of Chicago Press, National Bureau of Economic Research Conference Report, Chicago : Henderson, R. et Newell, G., 2011, URL : <http://www.nber.org/books/hend09-1>.

traire aux décisions des gestionnaires ou, du moins, ne pas tenir compte des contraintes imposées à l'entreprise car elles ne font pas partie du cœur de métier.

Ainsi, après la Seconde Guerre, les impératifs économiques aidant, les modèles d'organisation industriels étaient parfois en mesure de réserver de tels espaces de créativité en favorisant des groupes d'hommes de métier à exercer leur art tout en encadrant non pas leur travail mais les conditions d'exercice, et tout en pariant sur les futures opportunités en termes de production qu'un tel groupe était capable de fournir. Ce type d'organisation était-il pour autant entré dans les mœurs générales ? Comme nous allons le voir avec la firme IBM, il est certain que les secteurs industriels de hautes technologies avaient adopté ce modèle. Le fait de permettre à un salarié de consacrer la totalité ou une partie de son temps de travail à la réalisation de projets coopératifs était une manière d'envisager l'innovation.

Dans le cadre de la recherche universitaire, en revanche, il serait difficile d'établir une comparaison, dans la mesure où tout projet est par définition un *shunk work*, néanmoins, quelques similarités peuvent être établies notamment dans le cadre du projet Arpanet. Si, comme nous le verrons plus loin, nous concevons ce dernier comme une accumulation de technologies et d'améliorations par les utilisateurs, la coopération s'était établie sous la forme d'un encouragement général provenant à la fois des institutions participant au projet mais aussi par la cohésion du groupe de collaborateurs œuvrant pour différents programmes de recherche mais retrouvant dans ce projet de réseau les motivations de leur créativité.

Comme le montrent les *shunk works*, la représentation du travail d'ingénierie et de recherche dans le domaine des hautes technologies d'après-guerre incluait ces espaces de métiers dont les hackers des années 1960-1970 tels que les décrits Steven Levy¹ ne constituent qu'une forme, déjà ancienne, appliquée au domaine de l'informatique et tout particulièrement de la programmation. Dans ce dernier domaine, certaines entreprises avaient aussi développé leur propre modèle, ce qui fut en particulier le cas chez IBM.

IBM (International Business Machines) est une entreprise remarquable sur bien des points. Fondée en 1911 par Charles R. Flint, elle fusionna,

1. Steven LEVY, *Hackers. Heroes of the Computer Revolution*, New York : Dell Publishing, 1994.

sous le nom de Computing Tabulating Recording Company, les entreprises et productions qui constituaient sans doute l'avant-garde de la révolution technologique de la seconde moitié du XX^e siècle : la calculatrice de mesure de Julius Pitrap, l'enregistreur / pointeuse d'Alexander Dey et William Bundy et surtout la machine à statistiques à cartes perforées de Herman Hollerith¹. En cela, cette fusion préfigurait ce qui allait devenir une véritable culture d'entreprise chez IBM : rassembler les nouveautés et assembler les compétences². C'est sous la direction de Thomas J. Watson Jr., qui succéda à son père à la tête de l'entreprise en 1952 et jusqu'en 1971, qu'IBM s'offrit une place de choix dans l'industrie informatique, confirmée par l'apparition de l'IBM System/360 en 1964, une véritable gamme de machines, tournant sur le même logiciel avec les mêmes périphériques. Or, pour les mêmes raisons que dans l'industrie aéronautique, le secteur hautement concurrentiel des ordinateurs *mainframe* nécessitait des groupes de travail plus ou moins secrets³, dont l'objectif était de cristalliser les caractéristiques parfois anarchiques du processus de créativité, et rendre ce dernier *compatible* avec la structure commerciale de l'entreprise.

À la différence des *shunk works*, cependant, l'objectif voulu et décrit par T. Watson Jr.⁴ consistait à encadrer les travaux de groupes auto-formés, par affinité, identifiés parmi la population des ingénieurs-programmeurs

1. H. Hollerith, professeur de mécanographie au MIT à partir de 1882, fonda Tabulating Machine Co. en 1896, suite à son invention dont l'histoire technique est fort intéressante puisqu'il s'agit de ce qu'on appelle un transfert de technologie (entre le tissage industriel et ce qui deviendra l'informatique) qui fonda les bases de la mécanographie : l'utilisation du modèle de métier à tisser Jacquard, qui utilisait des rubans de cartonnages perforés, afin de mécaniser le traitement des données statistiques (notamment le pointage du recensement de population) en utilisant le principe des cartes perforées.

2. Voir l'ouvrage édité à l'occasion des 100 ans d'IBM, Kevin MANEY, Steve HAMM et Jeffrey O'BRIEN, *Making the World Work Better : The Ideas That Shaped a Century and a Company*, 1^{re} éd., IBM Press, juin 2011. Voir aussi le site internet consacré au centenaire de la firme : <http://www.ibm.com/ibm100>.

3. On pourra se référer à l'excellent article de Simon Donig sur la concurrence que se livrèrent les deux entreprises américaines IBM et CDC (Control Data Corporation) sur le marché de l'Allemagne de l'Est dans les années 1960. Cet article brise les idées reçues à propos de l'appropriation des technologies du bloc Ouest et montre à quel point la circulation des techniques suppose aussi une circulation des idées, nonobstant les questions d'espionnage industriel, inévitables dans cette histoire. Simon DONIG, « Appropriating American Technology in the 1960s : Cold War Politics and the GDR Computer Industry », dans : *IEEE Annals of the History of Computing* 32.2 (2010), p. 32–45.

4. Thomas J. WATSON, *A Business and Its Beliefs : The Ideas That Helped Build IBM*, 1^{re} éd., McGraw-Hill, avr. 2003.

en fonction de leur potentiel novateur. En d'autres termes, le rôle du manager était de capter l'apport théorique et technique des groupes de collaborateurs constitués d'esthètes de la programmation et de l'électronique, motivés par l'idée de « changer le monde ». Cette manière de manager la créativité en informatique était ce que T. Watson Jr. appelait « dresser les canards sauvages » (*wild ducks*), et cette expression perdura chez IBM tant elle représentait parfaitement la culture de l'entreprise, très proche du MIT et des groupes de hackers du monde universitaire :

Chez IBM, nous mentionnons souvent notre besoin d'avoir des *canards sauvages*. La morale est tirée d'un récit du philosophe danois Soren Kierkegaard, à propos d'un homme qui se met à nourrir les canards sauvages migrant vers le sud à chaque automne. Après quelque temps, certains de ces canards ne prirent plus la peine de voler vers le sud et passèrent l'hiver au Danemark, où l'homme les nourrissait. Puis ils volèrent de moins en moins. Après trois ou quatre années, ils devinrent si gros et paresseux qu'ils avaient toutes les peines du monde à voler. Kierkegaard attira l'attention sur ce point : vous pouvez apprivoiser des canards sauvages, mais vous ne rendrez jamais sauvages des canards apprivoisés. On pourrait ajouter aussi qu'un canard, une fois apprivoisé, n'ira plus jamais nulle part. Nous sommes convaincus que toute entreprise a besoin de ses canards sauvages. Et chez IBM, nous n'essayons pas des les apprivoiser.¹

Au début des années 1950, le besoin de machines était conséquent sur un marché au départ inexistant. Pour prendre l'exemple de l'IBM 701, sorti en 1952, 19 exemplaires furent fournis pour des universités, l'industrie aéronautique et les institutions gouvernementales. Pourtant, dès la sortie de l'IBM 701, les cadres commerciaux d'IBM s'aperçurent que les besoins concernaient à priori tous les domaines de l'ingénierie : du calcul des trajectoires de fusées à la conceptualisation d'infrastructures de transports, ou encore les bureaux gouvernementaux de statistiques et les banques, en somme tout ce qui nécessitait de lourds calculs sur un marché auparavant monopolisé par Univac, la première firme en mesure de vendre de telles machines (le premier UNIVERSAL Automatic Computer I fut produit en 1951). En produisant le supercalculateur IBM 701, en réponse à un appel

1. *ibid.*, p. 14.

d'offres de la Défense américaine, IBM remporta brillamment la première manche sur le marché et collabora avec la firme Engineering Research Associates (ERA), une entreprise auparavant subventionnée par l'U.S. Navy afin d'effectuer des recherches en cryptographie et qui était à l'origine de l'UNIVAC au sein de Remington Rand. Cette firme, qui s'était spécialisée en techniques de stockage d'information, avait largement amélioré le dispositif de mémoire à tambour magnétique, inventé au début des années 1930 par Gustav Tauschek, notamment en augmentant la densité et en réduisant le temps d'accès à la mémoire¹.

Grâce à l'appropriation de ces récentes technologies, IBM se trouvait en position de force pour s'avancer sur un marché plus étendu que celui de l'IBM 701. C'est à partir du moment où ce positionnement stratégique fut décidé que les *wild ducks* entrèrent en scène. L'objectif était de produire en série des ordinateurs utilisant l'électronique et le magnétisme, possédant une grande rapidité de calcul et des capacités de mémoire étendues, mais pour un coût abordable. En revanche personne chez IBM, et encore moins T. Watson Jr., n'était en mesure de savoir quels étaient exactement (techniquement) les besoins auxquels devaient répondre de telles machines, si ce n'est de permettre d'effectuer des calculs et stocker des informations. L'ingénieur de métier chez IBM devait s'accommoder d'une décision managériale qui n'intégrait pas les contraintes *client* de mise en œuvre. Si bien que ce qui fut assez longtemps reproché à IBM, à savoir un marketing agressif au détriment d'une véritable ambition d'équiper les clients avec des systèmes ultra-performants, fut exactement le résultat de cette volonté de couvrir un marché aux contours encore très flous.

En 1953, un an après l'arrivée de T. Watson Jr. à la tête de l'entreprise, l'IBM 650 (Magnetic Drum Calculator) fut mis sur le marché. Bien qu'il ne fit pas sensation du point de vue de ses capacités techniques, il fut le premier ordinateur vendu en masse (1800 exemplaires furent vendus ou loués en deux années de production), le moins cher du marché (un demi million de dollars contre un million pour la gamme Univac), et aussi le plus petit (capable de tenir dans une seule pièce).

1. Erwin TOMASH et Arnold A. COHEN, « The Birth of an ERA : Engineering Associates, Inc. 1946-1955 », dans : *Annals of the History of Computing* 1.2 (juin 1979), p. 83-97.

Alliance entre plusieurs technologies existantes et éprouvées par ailleurs, l'IBM 650 correspondait parfaitement à la culture IBM et à celle des *wild ducks*. En effet, même après les premiers exemplaires produits, les cibles marketing n'étaient pas des plus évidentes. IBM cherchant toujours à diviser ses cibles entre les entreprises et la recherche, il allait de soi que l'IBM 650, petit (proportionnellement aux autres machines sur le marché), modulaire, facile à programmer et permettant un accès rapide aux données (moins de 2,4 msec), passait devant les plus gros ordinateurs pour s'adapter à un usage dans le secteur industriel ou bancaire. Pourtant, ce furent les universités les plus demandeuses, pour des raisons à la fois commerciales et pratiques : une réduction importante leur était habituellement accordée par IBM en échange de cours sur la gestion de données ou en informatique générale ; par ailleurs, l'IBM 650 fournissant un des accès les plus rapides aux données, permettait par conséquent une plus grande rapidité dans la programmation et limitait les temps d'attente, un avantage clairement revendiqué par la population des programmeurs universitaires. Ainsi, IBM et ses *wild ducks* avaient fait un pari de développement basé non pas sur des cibles marketing ou sur un défi technologique, mais sur une certaine vision de la société : « on » avait forcément besoin de machines permettant un stockage optimal de l'information pour un coût raisonnable, ce qui en retour devrait se traduire par une expression plus précise des besoins. L'important n'était pas de savoir qui et pour quels usages, mais de développer en un maximum d'exemplaires des machines reposant sur les deux piliers de l'informatique moderne : rapidité et stockage. Tout l'aspect créatif du projet se situait dans l'amélioration des technologies existantes et dans une vision positive de l'utilité sociale de l'informatique.

Cela n'allait évidemment pas sans critiques, dont la principale concernait le choix de la technologie de mémoire utilisée dans l'IBM 650, à savoir une mémoire à tambour, alors même que les mémoires à tores (magnétique) de ferrite (*core memory*) étaient en passe de devenir le principal support des machines produites jusqu'aux années 1970. En effet, la même année 1953 où sortait sur le marché l'IBM 650, des systèmes à mémoire magnétique furent installés sur deux machines déjà assez anciennes : l'ENIAC et le Whirlwind. Concernant l'ENIAC (Electronic Numerical Integrator and Computer), conçu entre 1943 et 1946 à l'université

de Pennsylvanie par John P. Eckert et John W. Mauchly, il était destiné à effectuer des calculs balistiques suite à une demande militaire. L'utilisation d'un nouveau système de mémoire augmentait certes les capacités d'accès mais avait un impact bien moindre que sur le Whirlwind, conçu expressément en 1947 pour permettre un accès très rapide à la mémoire.

L'objectif du Whirlwind, conçu par Jay Forrester au Lincoln Laboratory (MIT), répondait là encore à une demande militaire consistant à créer un simulateur de vol, c'est-à-dire un système capable de traiter des données changeantes, d'où le besoin crucial de réduire le temps d'accès aux données. L'intégration d'un dispositif de mémoire à tores de ferrite fut une telle réussite que le Whirlwind fut surnommé Whirlwind II, puisqu'il avait plus que doublé sa vitesse de calcul (plus de 40.000 instructions par secondes). L'impact fut tel dans la communauté des programmeurs que les ingénieurs d'IBM n'attendirent pas la sortie de l'IBM 650 pour travailler sur ce dispositif et répondirent déjà en 1952 à l'appel d'offres du SAGE (Semi-Automatic Ground Environment – U.S. Air Force) et du Lincoln Laboratory pour produire un équipement hybride (mêlant mémoire à tores magnétiques et mémoire à tambours magnétiques) en une trentaine d'exemplaires¹. Cela était d'autant plus difficile pour IBM d'éviter les critiques affirmant que la firme préférerait développer et vendre plutôt que d'attendre un ou deux ans de développement supplémentaires et vendre une machine « grand public » en évitant les mémoires à tambours. On le devine aisément, ces critiques provenaient des chercheurs du MIT et d'Harvard qui, justement, avaient une visibilité assez complète des possibilités informatiques de la mémoire à tores magnétique, en particulier la non-volatilité de la mémoire (plus besoin d'une alimentation électrique spécifique ou d'un dispositif mécanique pour accéder à une mémoire). C'est dans le cadre de ses coopérations avec le Lincoln Laboratory et le SAGE que IBM pu ainsi garantir sa présence sur le marché des ordinateurs rapides, notamment face à Remington Rand (UNIVAC)².

Shunk works et *wild ducks* formaient donc des communautés d'ingénieurs dont l'organisation – basée sur un management global de groupes

1. Morton M. ASTRAHAN et John F. JACOBS, « History of the Design of the SAGE Computer - the AN/FSQ-7 », dans : *IEEE Annals of the History of Computing* 5.4 (1983), p. 343–344.

2. Emerson PUGH, *Memories that Shaped an Industry : Decisions Leading to IBM System/360*, Cambridge, MA : MIT Press, 1984.

non hiérarchisés mais motivés par un projet commun – était courante dans le secteur des hautes technologies, et entraînait la plupart du temps en interaction avec le monde académique. Les hackers du MIT des années 1960-1970 ne constituaient finalement qu’une résonance universitaire du management de la créativité d’ingénierie des firmes. Cette résonance s’explique bien entendu parce que nous nous situons dans un régime de coopération utilitaire entre ingénierie et recherche, entre les firmes et les universités. L’exemple d’IBM illustre bien, dès la création de cette entreprise, à quel point les liens avec le MIT étaient profonds et historiques (Herman Hollerith y enseignait la mécanographie), de même qu’avec tous les autres laboratoires d’informatique où IBM – tout comme DEC, Remington Rand, Burrough, et bien d’autres – allait puiser des enseignements et des apports théoriques en échange de tarifs préférentiels voire de dons de machines. Dans le cas du MIT, le positionnement stratégique d’IBM était aussi géographique : le Cambridge Research Lab, une succursale du T.J. Watson Research Center d’IBM, se trouve en fait à mi-chemin entre la banque Charles River et le campus central du MIT. C’est là que fut inventé en 1967 l’IBM 360/67, doté de mémoire virtuelle et fonctionnant en temps partagé avec TSS (Time-Sharing System), une adaptation directe du système CTSS/Multics (nous reviendrons plus loin sur ce système) pour les ordinateurs *mainframe* d’IBM. Mais cette résonance s’explique aussi parce que, comme les *wild ducks* d’IBM, les hackers du Tech Model Railroad Club et du MIT (pour ne prendre que cet exemple dont il va être question par la suite), étaient convaincus que l’informatique avait un impact positif sur la société par le biais de l’avancement des sciences : les sciences du langage, de la communication mais aussi la cybernétique naissante, en particulier grâce à la programmation et l’apparition des langages de programmation dont LISP est une illustration flamboyante d’abstraction logique et d’applications directes à l’ingénierie (ce qui fut concrétisé plus tard par la création de machines LISP).

Ces modèles d’organisation et d’innovation constituèrent un héritage à la fin des années 1960, proprement émulé par les interrelations entre l’industrie et la recherche universitaire. Cela donna lieu à la construction d’un archétype hacker, dont il faut dès à présent déterminer les contours pour comprendre le passage du groupe d’acteurs novateurs à l’idée de communautés de programmeurs.

3. Le travail hacker

Le célèbre livre de Steven Levy, *Hackers. Heroes of the Computer Revolution*, qui a connu jusqu'à aujourd'hui plusieurs rééditions, peut être considéré comme la première référence où le processus de cristallisation de l'« esprit hacker » se révèle de manière claire et documentée, à partir de nombreuses interviews¹. Que nous apprend cet ouvrage ? Écrit par un journaliste au début des années 1980, il obéit d'abord à un projet de constitution d'une mémoire des « pionniers » de l'informatique aux États-Unis, et en particulier l'identification d'une communauté de programmeurs dont les activités ont donné lieu à différents types d'innovations techniques dans la conception d'ordinateurs et de logiciels qui constituèrent une part significative de la « révolution informatique » de la fin des années 1950 au début des années 1980. L'analyse de Steven Lévy repose sur l'identification de plusieurs archétypes présents dans cette communauté de programmeurs se reconnaissant, d'après leurs propres propos, comme des hackers.

Le terme lui-même est assez vague et sa signification est compréhensible avant tout à partir du contexte où il est possible de parler d'une *culture* hacker, non pas en fonction des productions issues de leur secteur d'activités en ingénierie (l'informatique des deux décennies 1960 et 1970 est un domaine où se mêlent les spécialités en programmation, réseau et électronique), mais en fonction des pratiques qui rassemblent les hackers en un groupe structuré : l'éthique hacker. L'éthique dont il est question ici est à entendre comme un ensemble de principes régulateurs de l'action, une normalisation de l'activité professionnelle dont les racines tacitement reconnues par la communauté se retrouvent, selon S. Lévy, dans les activités du Tech Model Railroad Club (TMRC), le club de modélisme ferroviaire du MIT, en particulier durant les années 1958-1959, lorsque les membres chargés de l'élaboration des circuits électriques et des signaux avaient un accès, à des fins de calcul, au TX-0, le premier ordinateur à

1. Ces interviews relèvent de toute une tradition de l'histoire orale dans le domaine de l'histoire informatique. Pour s'en convaincre, il suffit de constater l'immense travail du Charles Babbage Institute dans la collecte de transcriptions d'interviews d'acteurs de l'histoire informatique et des ordinateurs, depuis la fin des années 1980. Beaucoup d'historiens, dont on peut lire les travaux dans les *Annals of the History of Computing*, ont travaillé sur ces archives inestimables.

transistor (Transistorized Experimental computer zero) issu du Lincoln Laboratory du MIT.

S. Lévy montre combien ce club joue, pour prendre une métaphore photographique, le rôle d'un révélateur de pratiques d'ingénierie basées exactement sur les principes managériaux de l'industrie (*wild ducks*) mais de manière auto-gérée : des bidouilleurs de talent à l'esprit potache qui trouvaient dans ce club un lieu en dehors des contraintes administratives et académiques pour exercer leurs talents dans une logique de projet et partager des connaissances et des ressources (le TX-0 en particulier, dont l'accès était limité). S. Lévy identifie six principes dans l'éthique hacker¹ :

1. L'accès aux ordinateurs – ainsi que tout ce qui peut permettre de comprendre comment le monde fonctionne – doit être universel (pour tous) et sans restrictions.
2. Toute information doit être libre.
3. Se méfier de l'autorité – promouvoir la décentralisation.
4. Les hackers doivent être jugés sur leurs activités (leurs hacks) et non suivant des critères « bidons » comme le diplôme, l'âge, l'origine ethnique ou le rang social.
5. On peut créer l'art et le beau à l'aide d'un ordinateur.
6. Les ordinateurs peuvent améliorer notre vie.

Ces principes ont été analysés par le philosophe finlandais Pekka Himanen dans un essai intitulé *L'éthique hacker et l'esprit de l'ère de l'information*². Pour Himanen, les principes hacker dépassent le seul cadre de l'informatique et formalisent une nouvelle éthique du travail dans une forme de relation passionnée qui s'oppose à la conception protestante à la base du capitalisme selon laquelle le travail est une finalité de l'existence, ainsi que l'exprimait Max Weber. Selon ce dernier en effet,

[Le travail] est caractéristique de *l'éthique sociale* de la culture capitaliste et joue en un certain sens pour elle un rôle constitutif. C'est une obligation dont l'individu se sent et doit se sentir investi à l'égard du contenu de son activité *professionnelle*, peu importe en

1. LEVY, *op. cit.*, p. 28 sq.

2. Pekka HIMANEN, *L'Éthique Hacker et l'Esprit de l'ère de l'information*, Paris : Exils, Éditions de l'Attribut, 2001.

particulier qu'une saisie naïve l'identifie à l'exploitation pure d'une force de travail ou à celle de possessions et de biens (d'un *capital*).¹

Selon Himanen, le hacker adopte d'abord une attitude et un engagement avant de considérer ses activités comme un métier dont le mobile principal n'est pas l'argent (bien que ce ne soit pas incompatible) mais la coopération directe, l'adhésion collective à un projet et l'indépendance par rapport à des instances (académiques, industrielles, etc.).

Il est frappant de voir combien les hackers eux-mêmes ont toujours adhéré aux principes énoncés par S. Lévy. Tous se reconnaissent dans cet énumération, et citent bien souvent S. Lévy.

Pekka Himanen la reprenant à son compte, c'est Linus Torvalds, concepteur du noyau de système d'exploitation Linux en 1989, qui préface l'ouvrage du philosophe (et il ne s'agit pas que d'une forme de solidarité nationale entre ces deux finlandais). Dans sa biographie même, co-écrite en 2001 avec le journaliste David Diamond², Torvalds se décrit comme un hacker, nonobstant le fait qu'il considère que le partage des informations relève plus d'une attitude pragmatique visant à favoriser l'innovation et la créativité que d'un ensemble de valeurs morales auxquelles il serait censé adhérer³. Quant à Sam Williams, un autre journaliste, lors de l'écriture d'une première version de la biographie de Richard Stallman, il reprend les termes mêmes de Steven Lévy décrivant Stallman comme « le dernier des (vrais) hackers ». Puis, dans la réécriture de cette même biographie, avec votre serviteur comme témoin, Richard Stallman enrichit même l'annexe située en fin d'ouvrage traitant de l'histoire et de l'éthique hacker et

1. Max WEBER, *L'Éthique protestante et l'Esprit du capitalisme*, Paris : Flammarion, 1999, p. 94 (cité par Himanen, op. cit. p. 27).

2. Linus TORVALD et David DIAMOND, *Il était une fois Linux : L'extraordinaire histoire d'une révolution accidentelle*, Paris : Osman Eyrolles Multimédia, 2001.

3. Cette position le rendit célèbre à maintes reprises, notamment parce que Torvalds la tient comme contradictoire, par interviews interposés, à la conception de Richard Stallman, pour lequel les principes des libertés logicielles ont une valeur morale, en plus de normer les comportements. Plus qu'une simple divergence de point de vue, les propos sont souvent véhéments, ainsi que le montre un interview de Torvalds en mai 2011, pour le site LinuxFr, à l'occasion de l'anniversaire des vingt ans du noyau Linux : « [...] Je méprise complètement les gens qui tentent de pousser la GPL [NdR : la Licence Publique Générale, initiée par R. M. Stallman] comme étant de type *éthique*. Je pense que c'est de la pure connerie. Pourquoi ? Parce que l'éthique, pour moi, c'est quelque chose de privé. Chaque fois que vous l'utilisez dans un argument pour dire pourquoi quelqu'un d'autre devrait faire un truc, alors vous n'adoptez plus une attitude éthique. Vous devenez juste une tête de con moralisatrice ». Voir <http://linuxfr.org/news/linux-torvalds-l-interview-anniversaire-des-20-ans-du-noyau>.

reprenant le premier chapitre consacré de Steven Levy. Pour terminer cette énumération, nous pouvons aussi citer Eric S. Raymond, qui, non content de se reconnaître en hacker, écrit lui-même une brève histoire du *hacking* intitulée *A Brief History of Hackerdom*¹.

La raison pour laquelle l'appartenance à la communauté hacker n'est pas qu'un sentiment mais une véritable revendication se rapporte aux deux premiers principes de Steven Levy : l'accessibilité à l'information de manière égalitaire, et le partage sans restriction de cette information. Ce sont là les principes fondamentaux de la coopération dans l'activité de programmation. Nous laisserons aux économistes la difficile question de savoir comment la coopération ouverte sans limiter la diffusion de l'information permet d'optimiser la dynamique de l'innovation et si elle est ou non plus efficace que les mécanismes classiques. À l'instar de Bob Bemer qui prônait pour les ingénieurs l'apprentissage de la programmation, nous pouvons dire que les principes hacker impliquent l'absence d'une quelconque séparation théorique entre le programmeur et l'utilisateur. Cela impliquait par exemple que dans le cas des ordinateurs à temps partagé, où il s'agissait de fournir un service (de temps de calcul) aux utilisateurs, ces derniers ne devaient en aucun cas voir leur accès au système limité par des droits d'accès : chacun devait pouvoir améliorer le système et partager ces améliorations². Une transcription technique de ces principes revient à dire que le code d'un programme doit être lisible et modifiable par tous. La transcription juridique de ces « libertés logicielles » (ou de l'éthique hacker) que Richard Stallman a initiée au début des années 1980 s'appelle la licence libre, et les programmes, des programmes libres (ou *open source*), par contraste avec toutes sortes de programmes ne permettant pas l'application des libertés définies par les hackers³.

1. Eric S. RAYMOND et al., « A Brief History of Hackerdom », dans : *Open Sources. Voices from the Open Source Revolution*, Cambridge, MA : O'Reilly, 1999, p. 19–29.

2. Témoignage de l'esprit potache des hackers, un petit jeu est entré dans le folklore, consistant à appliquer cette éthique de l'accès au machine en fracturant (avec élégance) les portes des bureaux des professeurs qui empêchaient l'accès à leur terminaux « personnels » reliés au réseau. Voir sur ce point Sam WILLIAMS, Richard STALLMAN et Christophe MASUTTI, *Richard Stallman et la révolution du logiciel libre. Une biographie autorisée*, Paris : Eyrolles, 2010, pp. 64–65.

3. Après le travail de Richard Stallman et Eben Moglen au début des années 1980 pour définir, dans le cadre du projet GNU, une licence de copyright logiciel respectueuse des droits et libertés des utilisateurs, les quatre libertés logicielles à la base de la Licence Publique Générale (GPL) et formulées par la Free Software Foundation sont : la liberté d'exé-

Certes, nous venons de mentionner trois grandes « stars » qui, rétrospectivement se retrouvent pleinement dans une communauté à laquelle, bien qu'ayant des avis divergents (par exemple la différence d'approche entre l'*open source* et le *free software*), ils appartiennent et contribuent. L'effet rétrospectif doit ici être mesuré. Pour cela on peut se référer au fichier informatique Jargon File, créé en 1975¹, qui constitue un glossaire du jargon utilisé dans les communautés hacker des deux laboratoires d'intelligence artificielle du MIT et de Stanford (MIT AI Lab. et Stanford AI Lab. – SAIL) ainsi que des autres communautés mobilisées autour des projets comme ARPANET ou MAC, des langages de programmation (comme LISP), ou même de firmes privées produisant des ordinateurs. En consultant les archives du Jargon File (mis à jour jusqu'à aujourd'hui), on peut remarquer que le terme hacker constitue une entrée intéressante, dès la première version du fichier. Elle n'a pas subi de grands changements jusqu'à la version 2.9.8 en janvier 1992 où apparaît seulement l'entrée *hacker ethic* qui fait explicitement référence, sans exclusivité toutefois, aux principes normatifs du projet GNU initié par Richard Stallman, en ces termes : « toute information doit être libre et tout contrôle privé de celle-ci est mauvais (c'est la philosophie du Projet GNU) ». Dans la version de 1975, par contre, ce sont essentiellement les activités concrètes du hacker qui sont mentionnées :

HACKER — Originally, someone who makes furniture with an axe.

1. n. A person who is good at programming quickly. Not everything a hacker produces is a hack.
2. An expert at a particular program, example: "A SAIL hacker".
3. A malicious or inquisitive meddler who tries to discover information by poking around. Hence "keyword hacker", "network hacker".

cuter le programme quel qu'en soit l'usage, la liberté d'étudier le (code du) programme et le modifier pour ses besoins, la liberté de distribuer des copies du programme (de manière commerciale ou non), la liberté d'améliorer le programme et distribuer ces améliorations pour en faire profiter la communauté.

1. <http://jargon-file.org/archive/>.

Cette définition semble devoir se rapprocher davantage de l'interprétation de P. Himanen que de celle de S. Lévy. En effet, si nous partons du principe que le Jargon File représente, à un moment donné de l'histoire, un témoignage de la représentation du hacker par lui-même, il est assez frappant de n'y trouver que peu de relation avec l'interprétation « philosophique » de Steven Lévy, surtout si l'on considère que l'entrée correspondant à l'éthique hacker apparaît plus de douze ans après la rédaction de son célèbre ouvrage. Ici, le talent, l'expertise et la persévérance des hackers semblent être des qualités davantage mentionnées lors d'un entretien d'embauche qu'en référence à un idéal de vie. Le hacker « fabrique des meubles avec une hache » (première phrase de l'entrée correspondante du Jargon File), il est capable de s'adapter à des situations difficiles où les ressources sont limitées et son activité de programmeur s'en trouve d'autant plus novatrice. Ce sont bien là les atouts recherchés par la plupart des entreprises impliquées dans le développement de programmes informatiques, la fabrication d'ordinateurs ou la gestion de ressources de calcul en réseau, qui fleurissaient (ou étendaient leurs activités) durant les années 1960-1970. Les hackers inauguraient donc davantage qu'une nouvelle forme d'ingénierie informatique, mais bel et bien un mode nouveau de rapport au travail, une ingénierie artisanale avec de nouvelles normes auxquelles ils s'identifiaient, telles que mentionnées par P. Himanen.

Il faut cependant souligner que ce dernier considère les hackers sous le spectre très large d'une histoire de plus de trente années de développement de logiciel libre. C'est en tout cas la manière dont il répond aux plus sceptiques. Deux économistes, J. Lerner et J. Tirole¹, ont analysé en 2002 l'impact de la dynamique de projets *open source* sur le modèle économique des entreprises impliquées. S'inspirant des travaux de l'économiste B. Hermalin² sur le leadership, ils se réfèrent eux aussi à Max Weber et sa sociologie³ mais pour montrer cette fois que les projets *open source* n'échappent pas au modèle classique de l'organisation sociale du travail : la confiance accordée au leader, l'harmonie entre les objectifs du leader et ceux des programmeurs au niveau individuel, l'absence de fac-

1. Josh LERNER et Jean TIROLE, « Some Simple Economics of Open Source », dans : *Journal of Industrial Economics* 52 (2002), p. 197-234, mentionné par P. Jollivet (voir infra).

2. Benjamin E. HERMALIN, « Towards an Economic Theory of Leadership : Leading by Example », dans : *American Economic Review* 88 (1998), p. 1188-1206.

3. Max WEBER, *Économie et Société*, Paris : Plon, 1971.

teurs politiques ou commerciaux dans la reconnaissance du leadership (on pense bien évidemment au quatrième principe évoqué par S. Lévy, à propos du jugement), l'acceptation à priori de toute critique de fond visant à améliorer le projet, toutes ces qualités se retrouvant dans la théorie de Weber et qui ne semblent pas aussi révolutionnaires que Himanen le prétend. Ce dernier précise néanmoins que de nouvelles notions apparaissent bel et bien dans une hiérarchie davantage horizontale que verticale des projets de développement où les figures de supérieurs et d'exécutants ont tendance à s'effacer, notamment au profit de la possibilité d'une destitution du leader. Comme le dit l'économiste Pascal Jollivet en commentant Himanen :

Pourtant, malgré les apparences, une différence fondamentale existe entre ces figures et celle du supérieur hiérarchique : *le statut d'autorité est ouvert à quiconque*¹. Ce qui est déterminant, c'est qu'une spécificité institutionnelle des projets en logiciel libre – nul ne peut se prévaloir d'avoir la propriété de biens logiciels produits dans le cadre de projet en licence libre – génère les conditions matérielles et sociales pour que cette autorité soit effectivement *ouverte* et *destituable*.²

L'appellation de hacker possède donc plusieurs acceptions et il est difficile de déterminer avec exactitude comment une communauté de hackers à pu naître et s'identifier en tant que telle, c'est-à-dire autour de principes tels que ceux énoncés par Steven Lévy, ou, plus généralement, autour d'une conception nouvelle du rapport au travail. Si, comme le montre le Jargon File, la communauté hacker se définit elle-même par un tel concept, les principes éthiques relèvent néanmoins d'une interprétation qui formalise rétrospectivement l'organisation du travail hacker, qui repose sur une division strictement horizontale du travail et une collaboration à tous les niveaux de la conception, de la programmation et de l'utilisation.

En effet, à la différence des journalistes écrivant les épisodes biographiques qui marquent l'histoire du logiciel libre, tel Steven Lévy, David Diamond ou Sam Williams, les historiens se sont contentés de considérer comme tangible l'identification de cette petite population de programmeurs de génie, tant au MIT qu'à Stanford, sur laquelle repose une partie

1. HIMANEN, *op. cit.*, p. 80.

2. Pascal JOLLIVET, « L'éthique hacker et l'esprit de l'ère de l'information de Pekka Himanen », dans : *Multitudes* 1.8 (2002), p. 161–170, p. 165.

significative de l'innovation informatique des années 1970. Ainsi, Paul Ceruzzi, spécialiste de l'histoire des ordinateurs modernes, adopte une attitude différente lorsqu'il s'agit de situer, contextualiser, les pratiques hackers. Par exemple, à propos du rôle qu'a joué l'entreprise DEC (Digital Equipment Corporation) en permettant aux programmeurs du MIT de modifier le PDP1 aimablement donné par Kenneth Olsen (directeur de DEC), Ceruzzi mentionne :

Olsen donna un autre PDP-1 au MIT où il devint la référence légendaire dans la culture des hackers plus tard commémorée par le folklore populaire.¹

Les termes et expressions comme « légendaire » et « folklore populaire » ne laissent aucun doute quant à la difficulté pour l'historien à prendre pour argent comptant l'interprétation de Steven Lévy pourtant accréditée par les hackers eux-mêmes, y compris ceux qui figurent parmi les acteurs principaux des décennies 1960 - 1970 - 1980. Leurs témoignages, bien que leurs recoupements permettent d'accréditer des faits, ne sont pas toujours les sources les plus fiables quant à l'interprétation de ceux-ci. Cependant, un consensus semble se dessiner dans l'acception du hacker comme un archétype d'agent vecteur de changement, à la fois par le partage de pratiques communes de travail, l'appartenance à une communauté ou à un projet social positif (même s'il n'est pas clairement défini) et par la place laissée à l'appropriation collective des connaissances nécessaires à l'innovation. Le hacker serait donc doublement vecteur de progrès : l'acteur du progrès technologique aux sources de la société de l'information d'aujourd'hui, mais aussi acteur de changement social, d'une nouvelle forme d'appropriation du changement technologique par le partage des connaissances et de l'immatériel (l'information en général, les logiciels, les pratiques d'ingénierie, etc.).

Il est souvent tentant de penser le changement technologique sur une échelle linéaire, que par commodité l'on nomme *progrès*. Cette idée n'est cependant que l'impression d'une forme d'interdépendance des technologies au fil du temps. Ainsi, d'un point de vue trivial, il aura bien fallu inventer la roue et le moteur à explosion pour inventer la voiture. Pour être exact, il n'est en fait ici question que de la valeur *cumulative* des tech-

1. CERUZZI, *op. cit.*, p. 128.

nologies¹ dans une société, ainsi que l'avait très justement analysé Lévy-Strauss (J. Schumpeter nous parle d'innovation *incrémentielle*). L'innovation s'analyse donc avec les notions de combinaison, mutation, transformation et appropriation, en opposition avec celle d'une prétendue continuité. Dans une société moderne, ces principes sont habituellement bien compris et toute entreprise confrontée à la nécessité concurrentielle dispose en elle les moyens d'encourager l'innovation, avec des méthodes plus ou moins efficaces d'alliance entre ingénierie, recherche et moyens humains.

L'histoire d'Internet illustre particulièrement bien l'*accumulation* des technologies à l'œuvre dans la formation d'un réseau de réseaux (réseaux d'acteurs / réseautage et réseaux informatiques). L'arrivée des technologies de réseau (avant qu'Internet ne soit devenu générique) marqua sans doute la dernière étape structurelle de la naissance d'une communauté hacker, ou plus généralement de la culture Libre.

4. Réseau

À la différence de projets clairement identifiés et stratégiquement planifiés, relevant parfois d'une volonté gouvernementale à l'image du projet Manhattan, Internet s'est développé de manière presque anarchique, avec un nombre d'utilisateurs restreints (les premiers utilisateurs de systèmes informatiques de communication à distance), des collaborateurs travaillant sur des aspects techniques différents, avec leurs calendriers propres de mise en œuvre, en somme, une accumulation de projets et de recherches théoriques, dont la convergence ne se comprend que d'un point de vue macroscopique et de manière rétrospective. Pour reprendre les termes de l'économiste Peter Meyer, il s'agirait donc d'une « invention collective », c'est-à-dire « un développement dont les améliorations et les découvertes expérimentales au sujet des processus de production ou les outils sont régulièrement partagés. »²

1. Bruno JACOMY, *L'Age du plip : Chroniques de l'innovation technique*, Paris : Seuil, 2002.

2. Philip MEYER, *Precision Journalism. A Reporter's Introduction to Social Science Method*, Indianapolis : Indiana University Press, 1973.

Longtemps considérée comme l'incubateur principal d'un « projet Internet », la DARPA¹ fut dotée de moyens considérables aux frontières des sciences informatiques. Des chercheurs issus de plusieurs nationalités vinrent se joindre aux équipes dont l'une des plus importantes fut le projet Arpanet (Advanced Research Projects Agency Network). De nombreux ouvrages² furent consacrés à cette histoire passionnante, mêlant la recherche académique, les institutions publiques et la planification d'État par les agences de moyens comme l'IPTO (Pentagon's Information Processing Techniques Office). Cette dernière, dédiée en premier lieu au développement de moyens pour la défense militaire, encouragea les interactions entre les institutions (entre l'ARPA et le monde académique) et le résultat fut la transition, effectuée sur une seule décennie, entre une informatique lourde et coûteuse (celle des énormes ordinateurs à traitement par lot, réservés à la recherche ou à quelques domaines bien spécifiques comme la banque ou les assurances) et une informatique réactive, avec une augmentation considérable de la puissance de calcul des ordinateurs, eux-mêmes dotés d'affichages graphiques, et assurant des systèmes de communication hautement sécurisés.

À la lecture des principales publications entre 1958 et le début des années 1970, les deux principaux concepts qui permirent cette transition furent le temps partagé (*timesharing*) et la commutation de paquet. Pour l'exprimer simplement, le temps partagé permettait une gestion de l'accès au temps de calcul via un système d'interruption : le traitement de l'information lancé par un utilisateur pouvait s'interrompre pour attendre une autre série d'instructions tandis qu'un autre processus, lancé par un autre utilisateur, pouvait se dérouler de manière à optimiser le temps d'utilisation des capacités de la machine. Les impacts furent de trois types : le développement concurrentiel de machines possédant de grandes vitesses de traitement, la possibilité d'augmenter le nombre d'utilisateurs simultanés sur une même machine, et, corrélativement, l'abaissement du coût du temps de calcul. La commutation de paquet, quant à elle, s'oppose au concept de circuit commuté telle la communication téléphonique qui or-

1. Defense Advanced Research Projects Agency. Agence créée en 1957, le D fut ajouté vers 1972.

2. Voir en particulier Arthur L. NORBERG et Judy E. O'NEILL, *Transforming Computer Technology : Information Processing for the Pentagon, 1962-1986*, Baltimore : The Johns Hopkins University Press, fév. 2000.

ganise la circulation de l'information de point à point par un seul chemin élaboré au besoin (« Mademoiselle, passez-moi le 22 à Anières »¹).

Faire circuler l'information par paquets² permettait l'utilisation de chemins multiples, à l'échelle de tout le territoire et avec des vitesses record ce qui, d'un point de vue militaire a fini par représenter un intérêt stratégique d'importance, bien que les activités de l'ARPA n'étaient pas exclusivement dédiées à la recherche militaire³. Le projet ARPANET fut donc un projet qui a d'abord permis l'alliance entre plusieurs technologies sur un modèle expérimental et en réunissant, entre 1967 et 1969, seulement quatre universités-cobayes (Stanford Research Institute, University of California Los Angeles et Santa Barbara, University of Utah), ainsi que la firme Bolt, Beranek & Newman (BBN) qui avait répondu à l'appel d'offres pour la maintenance du réseau et l'élaboration des routeurs (IMP – *Interface Message Processor*) sur la base des ordinateurs de la gamme PDP fournis par la firme Honeywell. À cela nous devons ajouter la rapide intégration du projet Multics et de ses acteurs au MIT.

L'idée *Internet* n'intervint qu'assez tard, d'abord sur l'idée de l'*internetting*, c'est-à-dire la possibilité d'utiliser les petits réseaux existants et hétérogènes en permettant leur interconnexion, d'autant plus facilitée par la possibilité de rendre les paquets d'information indépendants (datagrammes), de manière à favoriser leur passage de réseaux en réseaux. Pour cela, il a fallu développer collectivement tout un ensemble de protocoles de communications qui débouchèrent sur les protocoles TCP/IP. Ce n'est qu'en octobre 1972, lors de la première International Conference on Computer Communication qu'eut lieu une démonstration publique des principes de l'*internetting*⁴. Ce fut ensuite l'Amendement Mansfield qui, en 1973, circonscrit les recherches de l'ARPA au domaine relevant uniquement de la Défense (ARPA devint DARPA). La controverse qui s'ensuivit fut essentiellement alimentée par la National Science

1. Référence au sketch célèbre de Fernand Raynaud (1926-1973) à propos des relations difficiles entre un client et une préposée des Postes et Télécommunications.

2. Il faut encore que ces paquets puissent obéir à des protocoles de communication.

3. Alex ROLAND et Philipp SHIMAN, *Strategic Computing DARPA and the Quest for Machine Intelligence 1983-1993*, Cambridge, MA : MIT Press, 2002.

4. Vinton G. CERF, « Computer Networking : Global Infrastructure for the 21st Century », dans : *Computing Research : A National Investment for Leadership in the 21st Century*, Washington , DC : CRA, 1995, URL : <http://www.cs.washington.edu/homes/lazowska/cra/networks.html>.

Foundation car cette restriction n'était pas seulement institutionnelle mais privait l'ARPA des fonds conséquents apportés par celle-ci et de toutes les perspectives en termes de marchés que laissaient présager les avancées menées depuis plus de dix ans.

Pourtant, comme le note Shane Greenstein¹, l'Amendement Mansfield ne mit pas un terme aux échanges entre les acteurs provenant d'institutions différentes. Il y a deux raisons à cela, qui nous ramènent aux années 1960. La première est que ARPANET se devait de composer un réseau permettant l'émergence de protocoles communs, de manière à assurer la viabilité des échanges d'information non seulement entre machines mais aussi entre différents réseaux et, pour cela, avait besoin d'accéder à des réseaux autres que militaires. La seconde raison focalise sur les rôles individuels. Comme l'explique très bien Alexandre Serres², l'ARPA bénéficiait d'une assez large autonomie vis-à-vis de son agence de moyen, l'IPTO, grâce aux relations interpersonnelles entre acteurs des deux instances. D'un autre côté, la proximité entre l'ARPA et la firme BBN doit être soulignée. Cette firme, créée par deux professeurs du MIT (Leo Beranek et Richard Bolt, auxquels se rajouta un de leurs étudiants, Robert Newman) avait par ailleurs des relations privilégiées avec le MIT où les chercheurs, toujours en relation avec leurs anciens collègues, travaillaient sur différents projets de BBN, qu'il s'agisse de la mise en place de l'ARPANET à la conception de programmes adaptés aux ordinateurs PDP de chez Digital Equipment Corporation, elle aussi partenaire de BBN.

En somme, l'émergence d'ARPANET n'est pas seulement la création d'un réseau informatique, c'est aussi, par l'imbrication entre l'industrie, la recherche et les projets militaires, un réseau relationnel d'affinités³ : non seulement un nombre important de chercheurs et d'étudiants devinrent familiers avec les principes développés par l'ARPA, mais c'est

1. GREENSTEIN, *op. cit.*

2. Alexandre SERRES, « Aux sources d'internet : l'émergence d'ARPANET », Thèse en Sciences de l'information et de la communication, Rennes : Université de Rennes, 2000, p. 347 *sq.*

3. Les contre-méthodes hacker qui seront survolées plus loin ont même radicalisé cette tendance à transformer ARPANET en un système de réseautage social, notamment avec l'amélioration de systèmes de messagerie électronique, ce qui préfigura le devenir de l'Internet. On peut voir sur ce point Camille PALOQUE-BERGES, « Entre trivialité et culture : une histoire de l'Internet vernaculaire. Emergence et médiations d'un folklore de réseau », thèse de doct., Paris : Paris 8, 2011.

aussi pour l'ARPA la reconnaissance qu'une grande partie de sa force de travail, d'inspiration et de conception était externalisée sur les acteurs-utilisateurs, tous capables d'apporter leur lot d'améliorations et de mise en œuvre. En somme, ce que montre l'exemple de l'ARPA, c'est qu'une dynamique d'échanges et d'appropriations de concepts, d'expertise et de connaissances a été aménagée par les agences de moyens, IPTO et NSF, de manière à institutionnaliser les pratiques et identifier, sur une base de confiance mutuelle, les apports pertinents à un projet encore flou, qui devint plus tard Internet.

C'est tout à fait paradoxalement que cette institutionnalisation et ces relations interpersonnelles de convergences d'intérêts, donna le jour à d'autres communautés de pratiques qui entraient en conflit avec l'ordre établi bien qu'elles contribuèrent grandement aux structures informatiques en réseau.

Comme on peut le montrer en étudiant justement le cas des contributions au noyau Linux¹, dans un processus d'innovation, les individus qui créent de nouvelles connaissances produisent en même temps un apprentissage organisationnel, qui permet de cristalliser cette connaissance au niveau collectif, en formant une communauté épistémique. Cette cristallisation s'effectue à travers la circulation de « bonnes pratiques », c'est-à-dire celles qui favorisent la circulation des connaissances. L'innovation dépend donc essentiellement d'un cadre social auto-incitatif dont la première motivation est la mise en commun des connaissances.

Cette conclusion est à mettre en perspective avec ce qui se passa en 1969, lors de la jonction entre la phase de maturité d'un système d'exploitation à temps partagé (CTSS – Compatible Time-Sharing System / Multics) d'où découlait la notion de *computer utility*, c'est-à-dire l'avènement d'une économie de service terminal / ordinateur central, et l'expérimentation « universitaire-industrielle » de transfert de l'information en réseau (la naissance d'ARPANET, qui reliait alors plusieurs campus) rendue possible grâce aux machines de BBN et de DEC. Les retombées utilitaires de ces projets laissaient présager un marché intéressant comme, pour prendre

1. Patrick COHENDET, Frédéric CRÉPLET et Olivier DUPOUËT, « Innovation organisationnelle, communautés de pratique et communautés épistémiques : le cas de Linux », dans : *Revue française de gestion* 29.146 (2003), p. 99–121, URL : <http://www.cairn.info/revue-francaise-de-gestion-2003-5-page-99.htm>.

des exemples autres que militaires, équiper les banques de systèmes de gestion, permettre aux industries de mutualiser des gros calculateurs, voire de permettre à tous les habitants d'une ville d'accéder, via autant de terminaux par foyer, à des ressources système et même des contenus informatifs¹... Or, ce qui caractérisait à cette époque le système CTSS au MIT était la volonté des administrateurs à rendre étanches les sessions d'utilisation de l'ordinateur central, tout en instaurant une hiérarchie de droits d'utilisation, reflets structurels de la hiérarchie institutionnelle entre chercheurs, professeurs, directeurs, ingénieurs - programmeurs. Ce type de rigidité tendait à se mettre en place dans toutes les universités où s'expérimentaient les prémisses d'Internet, et allait à l'encontre des *bonnes pratiques* mises en œuvre dans la communauté hacker qui dès lors se reconnut en tant que communauté partageant un ensemble de valeurs différentes (les principes hacker de mises en commun des connaissances et de l'information sans contraintes) et qui s'exprimaient en réaction à l'organisation institutionnelle, par l'intermédiaire du réseau lui-même.

Entre 1968 et 1969, le témoignage le plus saisissant fut la création, par les hackers du MIT, du système ITS (Incompatible Timesharing System), nommé en réaction au système CTSS. Les concepteurs de l'ITS furent Richard Greenblatt, Tom Knight, Jack Holloway et Stuart Nelson. La principale caractéristique de l'ITS était la possibilité de transformer le système lui-même à chaque instant : connecté en permanence au réseau, l'ITS permettait à un utilisateur de travailler sur tous les fichiers (y compris les fichiers système) depuis n'importe quel terminal en toute transparence. Ce n'était pas anodin, puisque cette pratique donna naissance à deux autres innovations dont profitèrent les autres projets et bien des systèmes aujourd'hui : l'exploitation asynchrone des processus (plus besoin d'attendre la fin d'un processus pour en entamer un autre) et l'utilisation du PC-Lusering permettant de gérer stratégiquement les files d'attente en organisant une surveillance des processus en cours et qui entrent en concurrence pour l'acquisition de temps de calcul². En somme : une optimisation des

1. C'était la vision du sociologue Ted Nelson, inventeur de l'hypertexte, de la bibliothèque universelle informatique et du projet Xanadu.

2. Alors qu'avec CTSS, il fallait un superviseur pour répartir la charge de calcul, ce qui constituait bien sûr une porte ouverte à une hiérarchie d'utilisateurs (entre ceux dont les accès à la machine étaient plus importants en raison de recherche en cours et ceux dont les accès à la machine consistaient à créer des programmes pour le plaisir ou pour des objectifs

ressources du système et du temps de calcul disponible, pour un système informatique permettant de gérer les tâches de chacun.

Il faut lire entre les lignes de l'introduction du manuel de l'ITS¹, où l'on apprend que ce système a été élaboré « sur mesure » (*tailored*) pour le Projet MAC (le groupe du MIT travaillant en intelligence artificielle). Il a été conçu comme « une tentative de fournir des avantages potentiels d'un système à temps partagé », ce qui sous-entendait que CTSS ne le faisait pas. On peut énumérer par exemple, l'accès de tous les utilisateurs sans ordre hiérarchique (ni mot de passe) ou la possibilité pour tous de disposer d'une commande d'interruption du système. Bien entendu l'ITS ne s'adressait qu'à un « nombre limité d'utilisateurs » qui avaient « besoin d'un haut niveau de services sophistiqués ». En d'autres termes, l'ITS démontrait aussi que la créativité en matière de programmation ne nécessitait pas seulement l'accès de tous à des ressources matérielles, aussi performantes soient-elles grâce aux remontées entre utilisateurs (les universitaires) et les fabricants. L'un des facteurs de réussite était aussi dans le partage sans contrainte de l'information et des droits d'utilisation, ce qui émulait la coopération entre programmeurs et non l'esprit de compétitivité. Si nous acceptons cet anachronisme, nous pouvons dire que le système ITS est le premier système d'exploitation « libre », dans le sens où il traduit à la lettre les principes hacker.

5. Une contre-culture ?

Si les hackers sont bien ces *ingénieurs hétérogènes* dont nous parlait B. Pfaffenberger, porteurs de valeurs et de représentations, il est toujours étonnant d'apprendre sous la plume de nombreux auteurs du monde journalistique qu'ils sont considérés comme les acteurs d'un mouvement de contre-culture. Si l'on s'en tient à la définition de ce néologisme (*counter-culture*) créé par le sociologue-historien Théodore Roszak² analysant les

beaucoup trop abstraits pour les administrateurs). Voir sur ce point Simson GARFINKEL, *Architects of the Information Society : Thirty-Five Years of the Laboratory for Computer Science at Mit*, Cambridge, Mass. : MIT Press, 1999.

1. <ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-161A.pdf>.

2. Theodore ROSZAK, *The Making of a Counter Culture : Reflections on the Technocratic Society and Its Youthful Opposition*, Berkeley, CA : University of California Press, 1995.

mouvements de jeunesse des années 1960, la contre-culture est un mouvement de revendication contestataire à l'encontre des régimes technocratique des sociétés. De ce point de vue, les hackers des années 1960 sont bien loin d'être « contestataires » : ils passent le plus clair de leur temps à chercher les moyens de coopération les plus efficaces et, en cela, reproduisent des modèles de coopération déjà anciens hérités de l'ingénierie d'après-guerre. En revanche, ces modèles, ces « communautés de pratiques »¹, impliquent un ensemble de comportements incompatibles avec l'appropriation d'une technologie telle qu'elle permet au « propriétaire » d'utiliser un pouvoir sur les utilisateurs, en particulier lorsque ce pouvoir s'exerce sur les droits d'utilisation et instaure une hiérarchie là où la coopération se fait sur un mode égalitaire. Telle fut l'illustration de l'ITS.

L'émergence d'Internet fut à posteriori considérée comme une invention de chercheurs pour des chercheurs, c'est-à-dire une invention qui permettait de passer d'une informatique de calcul à une informatique permettant la circulation de l'information dans un cadre inédit et prometteur. Ce faisant, comme on le sait, Internet a multiplié les lieux de pouvoir (qu'ils soient démocratiques, comme la liberté d'opinion et d'information, ou moins démocratiques comme la surveillance des informations personnelles), et ce n'est que très récemment que la contre-culture hacker peut éventuellement se reconnaître comme telle.

Le hacking est un humanisme : il est passé d'une simple recherche d'efficacité dans l'organisation du travail à une éthique qui ne sépare pas utilisateur et concepteur. Et dans cette quête de l'optimisation des capacités cognitives et des savoirs-faire, ainsi que de la diffusion des connaissances et des créations, il est un modèle de bon usage des facultés humaines. Dans la mesure où, pour reprendre l'idée de Bob Bemer, il faudrait que tout utilisateur soit programmeur, le hacking constitue sans doute l'une

1. On se reportera à la définition qu'en donne le sociologue du travail E. Wenger, résumée par C. Paloque-Berges : « [La communauté de pratique] désigne les moyens mis en œuvre dans les processus d'apprentissage collectifs autour d'un objet d'intérêt commun par des groupes socio-professionnels auto-organisés. Ces processus se divisent en deux phénomènes : la participation (les modes de sociabilité engagés dans la pratique collective) et la réification (la production d'un artefact), qui sont en constante interaction, et redéfinissent l'identité et l'activité de la communauté ». Camille PALOQUE-BERGES, « La mémoire culturelle d'Internet : le folklore de Usenet », dans : *Le Temps des médias* 18.1 (2012), p. 111–123, URL : <http://www.cairn.info/resume.php?IDARTICLE=TDM0180111>, p. 121.

des clés des nouvelles formes d'innovations collectives de ce siècle, qu'il s'agisse de technologie, de culture ou de politique.

Références

- ABBATE, Janet, « L'histoire de l'Internet au prisme des STS », dans : *Le Temps des médias* 18.1 (2012), p. 170, URL : <http://www.cairn.info/revue-le-temps-des-medias-2012-1-p-170.htm>.
- ASTRAHAN, Morton M. et John F. JACOBS, « History of the Design of the SAGE Computer - the AN/FSQ-7 », dans : *IEEE Annals of the History of Computing* 5.4 (1983), p. 343–344.
- BEMER, Robert W., « What the Engineer Should Know Computer Programming », dans : *Automatic Control Magazine* 2, Feb./Mar. (1957).
- CERF, Vinton G., « Computer Networking : Global Infrastructure for the 21st Century », dans : *Computing Research : A National Investment for Leadership in the 21st Century*, Washington , DC : CRA, 1995, URL : <http://www.cs.washington.edu/homes/lazowska/cra/networks.html>.
- CERUZZI, Paul E., *A History of Modern Computing*, Cambridge, Mass. : MIT Press, 1998.
- COHENDET, Patrick, Frédéric CRÉPLET et Olivier DUPOUËT, « Innovation organisationnelle, communautés de pratique et communautés épistémiques : le cas de Linux », dans : *Revue française de gestion* 29.146 (2003), p. 99–121, URL : <http://www.cairn.info/revue-francaise-de-gestion-2003-5-page-99.htm>.
- DONIG, Simon, « Appropriating American Technology in the 1960s : Cold War Politics and the GDR Computer Industry », dans : *IEEE Annals of the History of Computing* 32.2 (2010), p. 32–45.
- GARFINKEL, Simson, *Architects of the Information Society : Thirty-Five Years of the Laboratory for Computer Science at Mit*, Cambridge, Mass. : MIT Press, 1999.
- GREENSTEIN, Shane, « Nurturing the Accumulation of Innovations : Lessons from the Internet », dans : *Accelerating Energy Innovation : Insights from Multiple Sectors*, Univ. of Chicago Press, National Bureau of Economic Research Conference Report, Chicago : Henderson, R. et Newell, G., 2011, URL : <http://www.nber.org/books/hend09-1>.

- HERMALIN, Benjamin E., « Towards an Economic Theory of Leadership : Leading by Example », dans : *American Economic Review* 88 (1998), p. 1188–1206.
- HIMANEN, Pekka, *L'Éthique Hacker et l'Esprit de l'ère de l'information*, Paris : Exils, Éditions de l'Attribut, 2001.
- JACOMY, Bruno, *L'Age du plip : Chroniques de l'innovation technique*, Paris : Seuil, 2002.
- JOLLIVET, Pascal, « L'éthique hacker et l'esprit de l'ère de l'information de Pekka Himanen », dans : *Multitudes* 1.8 (2002), p. 161–170.
- LERNER, Josh et Jean TIROLE, « Some Simple Economics of Open Source », dans : *Journal of Industrial Economics* 52 (2002), p. 197–234.
- LEVY, Steven, *Hackers. Heroes of the Computer Revolution*, New York : Dell Publishing, 1994.
- MANEY, Kevin, Steve HAMM et Jeffrey O'BRIEN, *Making the World Work Better : The Ideas That Shaped a Century and a Company*, 1^{re} éd., IBM Press, juin 2011.
- MCCARTHY, John, « History of LISP », dans : *ACM-SIGPLAN Notices* 13 (1978), 217–223.
- MEYER, Philip, *Precision Journalism. A Reporter's Introduction to Social Science Method*, Indianapolis : Indiana University Press, 1973.
- NORBERG, Arthur L. et Judy E. O'NEILL, *Transforming Computer Technology : Information Processing for the Pentagon, 1962-1986*, Baltimore : The Johns Hopkins University Press, fév. 2000.
- PALOQUE-BERGES, Camille, « Entre trivialité et culture : une histoire de l'Internet vernaculaire. Emergence et médiations d'un folklore de réseau », thèse de doct., Paris : Paris 8, 2011.
- « La mémoire culturelle d'Internet : le folklore de Usenet », dans : *Le Temps des médias* 18.1 (2012), p. 111–123, URL : <http://www.cairn.info/resume.php?IDARTICLE=TDM0180111>.
- PFÄFFENBERGER, Bryan, « The Social Meaning of the Personal Computer : or, Why the Personal Computer Revolution Was not a Revolution », dans : *Anthropological Quarterly* 61.1 (1988), p. 39–47.
- PUGH, Emerson, *Memories that Shaped an Industry : Decisions Leading to IBM System/360*, Cambridge, MA : MIT Press, 1984.

- RAYMOND, Eric S. et al., « A Brief History of Hackerdom », dans : *Open Sources. Voices from the Open Source Revolution*, Cambridge, MA : O'Reilly, 1999, p. 19–29.
- RICH, Ben R. et Leo JANOS, *Skunk Works : A Personal Memoir of My Years of Lockheed*, 1st Pbk. Ed, Back Bay Books, fév. 1996.
- ROLAND, Alex et Phlipp SHIMAN, *Strategic Computing DARPA and the Quest for Machine Intelligence 1983-1993*, Cambridge, MA : MIT Press, 2002.
- ROSZAK, Theodore, *The Making of a Counter Culture : Reflections on the Technocratic Society and Its Youthful Opposition*, Berkeley, CA : University of California Press, 1995.
- SERRES, Alexandre, « Aux sources d'internet : l'émergence d'ARPA-NET », Thèse en Sciences de l'information et de la communication, Rennes : Université de Rennes, 2000.
- STEELE JR., Guy L. et Richard P. GABRIEL, « The Evolution of Lisp », dans : *The second ACM SIGPLAN conference on History of programming languages*, HOPL-II, New York, NY, USA : ACM, 1993, 231–270, URL : <http://doi.acm.org/10.1145/154766.155373>.
- TOMASH, Erwin et Arnold A. COHEN, « The Birth of an ERA : Engineering Associates, Inc. 1946-1955 », dans : *Annals of the History of Computing* 1.2 (juin 1979), p. 83–97.
- TORVALD, Linus et David DIAMOND, *Il était une fois Linux : L'extraordinaire histoire d'une révolution accidentelle*, Paris : Osman Eyrolles Multimédia, 2001.
- WATSON, Thomas J., *A Business and Its Beliefs : The Ideas That Helped Build IBM*, 1^{re} éd., McGraw-Hill, avr. 2003.
- WEBER, Max, *L'Éthique protestante et l'Esprit du capitalisme*, Paris : Flammarion, 1999.
- *Économie et Société*, Paris : Plon, 1971.
- WILLIAMS, Sam, Richard STALLMAN et Christophe MASUTTI, *Richard Stallman et la révolution du logiciel libre. Une biographie autorisée*, Paris : Eyrolles, 2010.

Esquisse d'une archéologie de l'informatique communicante

Pascal ROBERT

Il est peut-être temps de tenter une « archéologie du savoir » informatique – et singulièrement de l'informatique communicante. Il y manquera, bien évidemment, le génie de Michel Foucault. Un Foucault qu'il n'est pas possible d'appliquer, mais dont il est néanmoins possible de s'inspirer – et de s'inspirer seulement¹. Cette archéologie ne doit pas forcément, du moins dans un premier temps, concerner le savoir technique de l'informatique ou l'informatique comme savoir technique. Elle doit d'abord, me semble-t-il, appréhender le socle des discours visionnaires sur lesquels s'est construite l'informatique communicante (à la fois conviviale et réseautique). Socle étrange, puisqu'il s'est révélé suffisamment solide pour que, sur cette base, s'élève l'essentiel de l'informatique graphique et de réseau. Socle également fragile, parce que, discursif et expérimental, il pouvait facilement s'effriter, voire se déliter et, si tel n'a pas été le cas, il faut reconnaître qu'il a subi des torsions, voire des trahisons – éventuellement

1. Foucault voulait passer sous les savoirs constitués afin d'en retrouver le terreau fondateur, entre dispositifs et discours pratiques. Non pas faire l'histoire de la médecine, donc, mais restituer la pratique de l'examen à la fois comme discours et comme dispositif d'observation qui, progressivement, va installer le socle sur lequel se construira un savoir à proprement parler médical. Il s'agit de revenir, ici, aux dispositifs et aux discours qui sous-tendent l'avènement de l'informatique communicante.

productives, cela dit. Socle qui, enfin, a produit un effet d'*ouverture* sur une autre manière de concevoir l'informatique comme dispositif, au-delà des grands systèmes propriétaires, entre micro-informatique et réseaux. Un principe d'ouverture dont a profité la logique du logiciel libre.

Or, le noyau dur de ce socle s'articule avec les travaux de Douglas Engelbart et Joseph Carl Robnett Licklider. Leurs visions, entre les années 1960 et les années 1970, vont être travaillées par une tension forte entre la logique de l'*apprentissage* et celle de la *convivialité* chez Engelbart et celles de la *modélisation* et de la *communication* chez Licklider. Ce sont les *quatre piliers* de notre informatique graphique et communicante. Son histoire est celle de leurs tensions, c'est aussi, quelque part, celle du Libre : on aurait pu imaginer une informatique qui promeuve largement l'apprentissage et la modélisation, ce à quoi s'attache plus volontiers le Libre, mais la micro-informatique et les réseaux ont préféré massivement survaloriser la convivialité et la communication ; les figures de l'avenir vont, peut être, se tracer entre apprentissage et communication et/ou convivialité et modélisation, à moins que l'on ne revienne aux positions d'Engelbart et de Licklider. . . . Quoi qu'il en soit, il nous faut passer *sous* cette histoire, ce en quoi consiste toute archéologie, afin de mieux comprendre, d'une part, les conditions d'émergence discursives et pratiques de ces quatre piliers et, d'autre part, le basculement en faveur de la convivialité et de la communication.

1. L'augmentation et le *bootstrapping*

L'apprentissage chez D. Engelbart renvoie à ce qu'il appelle le « *bootstrapping* », un mécanisme de production réciproque entre l'homme et le dispositif technique. « L'augmentation » des capacités intellectuelles que recherche D. Engelbart passe par la création – et le détour donc – d'un dispositif technique complexe (On Line System, NLS) qui exige un effort d'apprentissage¹. À travers cet effort, l'homme apprend (à développer le système et à utiliser le système) et par là même se transforme. Et cet homme transformé va à son tour modifier le système, etc. S'il y a de

1. Ce projet a été développé au sein d'un laboratoire justement appelé Augmentation Research Center (ARC) créé par Engelbart en 1968 dans le cadre du Stanford Research Institute.

la convivialité parfois là-dedans, elle est au service de ce mécanisme, elle n'est jamais autonome et ne constitue pas un objectif. C'est avec les travaux d'A. Kay notamment au PARC (Palo Alto research center) de Xerox dans les années 1970 que le « user friendly » va s'imposer comme un but. Ce que D. Engelbart vivra comme une sorte de trahison.

1.1 L'augmentation : le projet

Aujourd'hui le terme « augmentation » renvoie de manière privilégiée à une des modalités de conception et de manifestation des mondes virtuels 3D : il s'agit d'ajouter une couche de virtuel à la réalité qui, ainsi, s'en trouve justement augmentée. Mais le terme d'augmentation a d'abord été utilisé par Douglas Engelbart dans ses travaux des années 1960 pour désigner une réponse possible, une réponse technique en l'occurrence (et non organisationnelle par exemple), au défi de la gestion de la complexité.

1.1.1 Le défi de la complexité

La vision de Douglas Engelbart s'articule à une forte sensibilité avec la question de la montée en puissance de la complexité lors des années 1950 et 1960. Ainsi, Douglas Engelbart, dans un exercice d'histoire orale rétrospective en vient à souligner que :

I remembered reading about the people that would go in and lick malaria in an area, and then the population would grow so fast and the people didn't take care of the ecology, and so pretty soon they were starving again, because they not only couldn't feed themselves, but the soil was eroding so fast that the productivity of the land was going to go down. So it's a case that the side effects didn't produce what you thought the direct benefits would. I began to realize it's a very complex world. If you can pick a target that if you succeed will indeed produce the benefit you wanted, or it might also have negative side effects that are going to counteract the benefit.¹

La complexité, mal maîtrisée, entraîne des effets pervers, c'est-à-dire que même une intervention pleine de bonne volonté peut aboutir à un effet négatif qui compense et souvent dépasse l'effet bénéfique escompté *a*

1. Thierry Bardini, *Bootstrapping: Douglas Engelbart, Coevolution, and the Origins of Personal Computing*, Stanford: Stanford University Press, 2001, p. 10.

priori. Nous ne semblons pas suffisamment armés pour faire face à ces effets qui s'entrecroisent, s'enchevêtrent, se bouclent et se relancent. La réponse apportée à ce constat par Douglas Engelbart aurait pu n'être que théorique. Elle fut double, à la fois théorique, avec ce qu'il appellera le *bootstrapping*, et pratique, à travers l'invention d'un artefact technique. Or, ici la théorie n'est pas indépendante du dispositif lui-même, puisqu'elle est avant toute chose la théorisation de la manière de l'utiliser. Pourquoi une telle réponse ?

La question de la complexité n'est pas la seule qui inquiète Douglas Engelbart, car précise-t-il :

I began to realize the probability of your achieving your goal isn't terribly high, and the probability if you do achieve it that it's a success is low. So, you'd better start learning about that. Someplace along there, I just had this flash that, hey, what that really says is that the complexity of a lot of the problems and the means for solving them are just getting to be too much. The time available for solving a lot of the problems is getting shorter and shorter. So the urgency goes up. So then I put it together that the product of these two factors, complexity and urgency, are the measure for human organizations or institutions. The complexity/urgency factor had transcended what humans can cope with. It suddenly flashed that if you could do something to improve human capability to deal with that, then you'd really contribute something basic.¹

La question de l'urgence n'est autre que celle de la vitesse. En effet, plus la complexité croît et plus la réponse que les hommes peuvent lui apporter doit pouvoir être à la hauteur, parvenant à maîtriser un nombre toujours plus grand de facteurs eux-mêmes en variations, ce qui nécessite une puissance de traitement de l'information toujours plus grande et plus rapide. Il est clair à ses yeux que la seule intelligence de l'homme ne pourra pas répondre, seule, à ce défi. Autrement dit, notre monde devient à ce point complexe que notre culture doit se créer de nouveaux outils susceptibles de l'affronter. De nouvelles armes.

1. *ibid.*, p. 10.

1.1.2 L'augmentation comme réponse : le détour

La complexité ne peut pas être appréhendée à l'aide de notre seul cerveau. Mais il ne s'agit pas pour autant de renier ce cerveau. Il ne s'agit pas de lui substituer un autre cerveau, artificiel, qui viendrait suppléer à ses carences. C'est pourquoi notre intelligence doit, non pas faire l'objet d'un reniement au bénéfice de la machine comme le prétend le programme de l'intelligence artificielle (IA), mais faire l'objet d'une « augmentation », c'est-à-dire d'une amplification de ses propriétés et aptitudes. Ce qui exige de mettre au point un dispositif – et la philosophie qui va avec – qui soit capable de travailler avec l'être humain dans une relation qui, nous le verrons, les implique mutuellement.

L'augmentation constitue une réponse plutôt paradoxale à la question de la montée en puissance inexorable de la complexité des problèmes du monde. En effet, si la complexité est multipliée par l'urgence, alors la réponse exigée devrait elle-même relever de cette logique de l'urgence, autrement dit, être rapidement mise au point et diffusée. L'urgence même de la question devrait commander l'urgence même de la réponse. Or, l'augmentation correspond à un détour, à une réponse non pas directe, mais indirecte, une réponse qui, quelque part, prend le temps de sa construction. Car c'est cette prise en compte du temps et du processus même de sa construction comme devant faire partie intégrante de cette construction – et non comme un facteur externe à maîtriser comme une contrainte – qui fait toute l'originalité de la proposition de Douglas Engelbart. La réponse n'est pas donnée mais à élaborer, et cette réponse elle-même ne sera véritablement efficace que si l'on prend le temps d'apprendre à mettre au point un dispositif technique lui-même complexe, en ce sens qu'il n'est pas forcément simple d'accès ni de s'en servir, pour l'améliorer afin qu'il puisse affronter une complexité croissante. Autrement dit, semble-t-il (et c'est en tout cas l'analyse que j'en fais), même sa mise au point ne parviendra pas à la réduire, sans compter l'effet pervers qui, si l'on suit le raisonnement de Douglas Engelbart lui-même tel qu'il le présente plus haut, pourrait l'amener à l'accroître.

1.2 Le *bootstrapping* : le processus et l'outil

Ce mélange d'un dispositif technique et d'une philosophie à la fois de conception et d'apprentissage, c'est exactement ce que Douglas Engelbart désigne par le terme de *bootstrapping*. Douglas Engelbart va le mettre au service de la réalisation d'un artefact susceptible de participer à la maîtrise de la complexité en maîtrisant plus de données et plus rapidement sur cette complexité.

1.2.1 Le processus : l'effort comme principe

Le *bootstrapping* est un processus d'apprentissage qui ne vise pas à faciliter directement la manipulation d'un grand volume de données. C'est-à-dire que cette facilitation n'est pas un objectif en soi, c'est un moyen. L'objectif, c'est l'apprentissage, le processus d'apprentissage lui-même. Et ce processus ne passe pas obligatoirement par une logique de la facilité d'accès. Au contraire, non seulement l'effort d'apprentissage n'est pas banni mais il est même valorisé, car il est effectivement une valeur profonde de la philosophie du système. Autrement dit, il s'agit de mettre au point un dispositif technique qui est une réponse possible au défi que pose la complexité mais qui ne peut lui-même être trop simple, car il ne pourrait tout simplement pas absorber cette complexité. Nous l'avons vu, il en va d'une sorte de paradoxe : plus la complexité est importante, plus elle exige un outil complexe pour l'appréhender, lequel outil nécessite du temps pour sa mise au point et la possibilité d'apprendre à s'en servir de manière efficace. Donc l'outil ne peut être simple, ni son apprentissage. Au contraire, il doit impérativement avoir atteint un certain niveau de complexité pour se révéler performant et son apprentissage doit posséder un certain niveau de complexité pour être également efficace. Il faut de toute manière un minimum de temps, aussi rapide soit le fonctionnement du système et pertinent son apprentissage, pour atteindre ce niveau de maturité. Le concept de *bootstrapping* entérine et synthétise ces conditions.

Cela signifie que le dispositif technique recherché est justement moins mis au point qu'en recherche perpétuelle de sa propre complexification pertinente en tant qu'outil de gestion de la complexité... à commencer par la gestion de sa propre complexité. C'est donc un dispositif technique qui

ne peut être séparé, découplé de son processus d'apprentissage. Ce dernier n'est pas une contrainte, un contre-temps, un détour dommageable, il est un constituant intrinsèque du dispositif. Ce qui explique qu'il ne s'agit en aucune manière ici de simplifier pour simplifier, ou plutôt qu'il ne s'agit pas de rendre les choses plus faciles parce que plus intuitives. Prenons l'exemple du *Chord Keyset*. Il s'agit d'un outil d'interface fonctionnellement similaire à la souris, mais très différent dans sa configuration. Il s'inspire des touches du piano comme moyen de la médiation. Après tout, le piano fonctionne effectivement. Simplement, jouer du piano n'est en rien évident, cela requiert un apprentissage, un long et difficile apprentissage. Le *Chord Keyset* est bien évidemment beaucoup plus simple qu'un piano, puisqu'il ne comporte que cinq touches. Mais il n'est en rien un outil de la convivialité, car il n'est pas évident à manipuler. Il demande un *effort* d'apprentissage. Et selon Douglas Engelbart, qui a fourni cet effort, il est dès lors un outil encore plus efficace que la souris.

1.2.2 L'outil (NLS) : la convivialité comme imprévu et comme effet

Si Douglas Engelbart et Bill English en viennent à inventer la souris, ce n'est pas parce que celle-ci serait le résultat d'un effort volontaire de recherche de simplicité ou de facilité d'usage de l'interface. C'est surtout une question d'opportunité. En effet, la souris est d'abord née du besoin de différenciation d'un des commanditaires du laboratoire de Douglas Engelbart : la NASA. Il lui fallait que ses financements servent à quelque chose, mais surtout à quelque chose d'autre que ce à quoi servaient les financements de la Rand Corporation¹ notamment. Or, la Rand était en train de mettre au point ce que ses inventeurs (Ellis et Davis en 1964) ont appelé la *Rand Tablet*, un outil d'interface qui utilisait un *light pen*, un stylet, qui dessinait sur une surface dédiée et dont le dessin s'affichait simultanément sur un écran... Il revenait à Douglas Engelbart et son équipe d'inventer autre chose. D'autant plus que si, dans un tout premier temps, Douglas Engelbart pensait utiliser une *Rand Tablet* comme outil d'interface, la Rand Corporation le lui refusa.

1. La Rand Corporation était un *think tank*, un outil de réflexion et de financement de projets de recherche créé à la fin des années 1940 par l'US Air Force (1946). Elle deviendra ensuite (à partir de 1948) une *non profit organization* néanmoins encore largement financée par l'USAF. Elle sera l'un des principaux lieux de l'innovation stratégique et technologique au service de la défense américaine dans les années 1950.

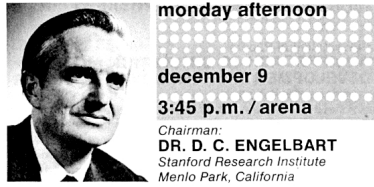
La souris vient d'une double différenciation : d'une part, alors que le *light pen* et la *Rand Tablet* permettent de tracer des graphiques, la souris fonctionne d'abord comme un pointeur qui permet soit d'actionner une opération afin d'écrire des mots, soit une opération pour tracer des traits ; d'autre part la souris permet une parfaite coordination entre ce que fait la main et ce que voit l'œil, alors que ce qui est actif est sur l'écran : la surface de la *Rand Tablet* est sensible et émet le signal, alors que c'est la souris qui le transmet, quelle que soit la surface ; elle est donc tout terrain, ce n'est pas ce que l'on trace sur la surface (écran ou plateau) qui compte, mais la coordination entre l'œil et la main, car on regarde l'écran alors que la main trace, autrement dit, on ne regarde pas la main mais le résultat de son action.

La création de la souris renvoie en quelque sorte aux effets latéraux de la gestion de la complexité dont parlait – mais autrement – Douglas Engelbart : elle n'est pas l'effet de ce qui est voulu et recherché – l'effort – mais de ce qui va à l'encontre de cette philosophie, puisqu'elle est le premier outil d'interface convivial. Il y a une sorte de paradoxe ironique dans la mise au point de la souris. En même temps elle peut être, à un autre niveau, considérée comme une expression possible du *bootstrapping* puisqu'elle provient d'un apprentissage du système en quoi consiste le laboratoire pris globalement.

En 1968, Douglas Engelbart fait une « démo » de son système NLS à la *ACM/IEEE Computer Society "Fall Joint Computer Conference"*, à San Francisco. Cette « démo » est disponible et visible sur internet ¹.

On y voit en effet Douglas Engelbart montrer son système tout en l'utilisant. La « démo », en ce sens, est cohérente intellectuellement et pratiquement avec la philosophie du *bootstrapping*, car elle offre une vue de ce que peut ce dispositif à travers le dispositif lui-même. Ce n'est pas un discours sur ce qu'il est possible de faire avec cet outil, c'est une mise en pratique devant les participants à une manifestation professionnelle de ce qu'il est possible de faire (à distance d'ailleurs, puisque si les outils d'interface – dont la souris – sont à San Francisco, l'ordinateur est à Stanford). Autrement dit, Douglas Engelbart prend le *risque* de faire cette « démo »

1. <http://sloan.stanford.edu/mousesite/1968Demo.html>.



a research center for augmenting human intellect

This session is entirely devoted to a presentation by Dr. Engelbart on a computer-based, interactive, multiconsole display system which is being developed at Stanford Research Institute under the sponsorship of ARPA, NASA and RADG. The system is being used as an experimental laboratory for investigating principles by which interactive computer aids can augment intellectual capability. The techniques which are being described will, themselves, be used to augment the presentation.

The session will use an on-line, closed circuit television hook-up to the SRI computing system in Menlo Park. Following the presentation remote terminals to the system, in operation, may be viewed during the remainder of the conference in a special room set aside for that purpose.

Annonce de la démonstration de D. Engelbart en 1968

en direct et inaugure ainsi le genre ¹. Mais ce risque, comme l'effort d'apprentissage, ne font-ils pas partie intégrante d'une certaine manière de penser et de pratiquer le *bootstrapping* qui est intrinsèquement innovante ?

La démo est intéressante par sa mise en scène. Car le dispositif adopté permet de voir à travers l'écran sur lequel les instructions et les données qu'introduit Douglas Engelbart s'inscrivent. Nous sommes de l'autre côté du miroir. Nous voyons ainsi Douglas Engelbart parler et nous le voyons écrire, bref, nous le voyons faire la démo comme les gens dans la salle l'ont vu ². Parfois nous avons une vue globale qui présente le système technique : écran et plateau disposés devant Douglas Engelbart avec les trois outils d'interface que sont le *Chord Keyset*, le clavier et la souris. Cette « démo » montre comment un homme, armé de ces trois outils peut créer des *statements* et jouer avec eux ; comment il peut les manipuler : les créer, les déplacer, les multiplier, les trier, les classer, etc. Bref, pour la

1. La « démo » est en effet un lieu qui donne à voir la technique à la fois dans son dispositif (que l'on fait fonctionner) et dans un discours pratique qui explique comment cela fonctionne : ce geste de D. Engelbart, invente ainsi une mise en scène de l'informatique qui est devenue une pratique récurrente dans le milieu.

2. Comme le souligne Engelbart lui-même : « *I saw the same image on my workstation screen there as was projected for the audience to see* », *ibid.*, p. 140

première fois on peut assister au prélude de ce qui va devenir ultérieurement notre quotidien, ce travail avec et sur des éléments, des données textuelles, chiffrées ou graphiques. Certes, Douglas Engelbart s'est entraîné et il pratique l'exercice avec une certaine aisance, mais à un moment où tout ce qui est devenu évident pour nous ne l'était pas, même pour lui (il lui arrive de se tromper).

Ne regardons pas cette « démo » avec nos yeux, notre cerveau et nos mains d'aujourd'hui. Astreignons-nous à la regarder avec les yeux de celui qui, à l'époque, utilisait une tout autre informatique. Cette évidence, qui est la nôtre, parce que nous reconnaissons là ce que nous avons appris à faire depuis, est tout simplement sidérante, radicalement nouvelle (sauf pour une extrême minorité de chercheurs). Regarder cette « démo », c'est remonter à ce moment de l'orée de la convivialité qui n'est encore en rien conviviale parce que beaucoup trop nouvelle. Quand bien même on éprouve inévitablement le sentiment qu'il y a là quelque chose qui change considérablement le rapport à l'ordinateur, rien ne dit encore que cette manière de dialoguer avec lui va devenir dominante. Elle n'est évidente à nos yeux qu'à oublier les apprentissages que nous avons tellement intégrés qu'ils nous habitent et se font oublier comme tels. Une part de la convivialité s'invente là, mais elle ne le sait pas (ni même ses inventeurs peut-être) et pourtant elle reste à inventer en tant que telle. C'est ce à quoi va travailler Alan Kay au Xerox Parc.

1.3 Torsion : A. Kay, la convivialité comme *a priori*

Alan Kay n'a jamais directement travaillé avec Douglas Engelbart. Mais à partir de 1971, une partie de l'équipe du laboratoire de Douglas Engelbart, l'ARC (Augmentation Research Center) au SRI (Stanford Research Institute), rejoint le Palo Alto Research Center de Xerox, grand lieu de l'innovation technologique en matière informatique des années 1970. Ce fut le cas de B. English, le co-inventeur de la souris.

Alan Kay connaît bien les travaux de Douglas Engelbart et de son équipe. Suffisamment pour que Douglas Engelbart se sente quelque peu plagié. En effet, souligne-t-il : « *they take parts of the architecture, they take the mouse, they take the idea of the windows, all of that, but the rest of*

NLS they just rejected »¹. Ce qui est resté sur le bord de la route, c'est justement le *bootstrapping*, la co-évolution par l'apprentissage réciproque : autrement dit on a conservé le *hard* et le *soft* mais pas la philosophie qui allait avec. Et, d'une certaine manière, aux yeux de Douglas Engelbart, on a tout changé. Ce n'est pas tant que les mêmes outils, plongés dans une approche différente auraient donné de nouveaux outils eux-mêmes différents. C'est que ces outils proposaient des propriétés dont l'expression va se trouver transformée par le changement de philosophie. Car ces propriétés étaient en quelque sorte canalisées (certains diraient bridées) par le système NLS alors qu'elles vont exploser si l'on peut dire hors de ses contraintes. De facilitateurs (les fenêtres comme la souris par exemple) au service de la gestion d'un outil de gestion de la complexité qui restait lui-même globalement plutôt complexe et requérait ainsi un apprentissage, ils deviennent des outils de la convivialité qui s'autonomisent en quelque sorte et deviennent ce autour de quoi le reste doit tourner, ils deviennent une fin en soi.

Car la philosophie de Kay est radicalement différente de celle de Douglas Engelbart. Le premier utilisateur auquel il pense pour son *Dynabook*² n'est en rien un informaticien de haut niveau. Ce n'est pas un chercheur ou un professionnel dans une entreprise, mais... un enfant. Le système doit être manipulable par un enfant, c'est bien pourquoi il doit être intuitif et convivial. Autrement dit, et quelque peu paradoxalement, la mise en convivialité doit tout faire tourner autour de ses outils (en amont, lors de la conception) afin qu'ils se fassent le plus oublier possible lors de l'usage de la machine. Car cet usage doit être transparent et l'apprentissage tout aussi indolore. C'est-à-dire se faire oublier comme apprentissage. Nous sommes à des années-lumière d'un processus qui reposait, lui, sur l'effort. Dès lors, les outils qui visaient dans ce cadre, non pas à éliminer l'effort, mais à lui permettre d'aller plus loin dans son efficacité, non pas à l'éradiquer, mais seulement à le déplacer pour encore mieux gérer

1. *ibid.*, p. 154.

2. « Several years ago, we crystallized our dreams into a design idea for a personal dynamic medium the size of a notebook (the *Dynabook*) which could be owned by everyone and could have the power to handle virtually all of its owner's information-related needs ». Alan KAY et Adele GOLDBERG, « Personal dynamic media », dans : *The New Media Reader*, sous la dir. de Noah WARDRIE-FRUIIN et Nick MONTFORT, Cambridge Mass. : MIT Press, 2003, p. 391-404.

la complexité, deviennent des arguments au service de la valeur centrale de la convivialité.

Parce que la figure de l'enfant ne passe pas forcément facilement en ce début des années 1970, même chez Xerox, l'étape ultérieure aura un référent certes moins radical, mais néanmoins encore bien naïf ou à tout le moins candide : une secrétaire, Sally, la cinquantaine bien avancée, comme le précisaient Tim Mott et Larry Tesler¹. Une secrétaire et non un technicien ou un ingénieur. Une secrétaire au service de la gestion d'une entreprise et dont le travail est un travail de fabrication et de manipulation de documents (lettres, factures, etc.). Larry Tesler, chercheur au PARC, passera ensuite chez Apple, emportant avec lui ses compétences dans le *drag-and-drop* et sa connaissance du nouveau modèle de la souris qui ne fonctionne plus avec des roues (modèle des années 1960 de D. Engelbart et B. English) mais avec une boule (modèle des années 1970 de Xerox). N'oublions pas que Xerox se pense d'abord comme une entreprise de services documentaires, une *document company*. L'informatique n'est pas une fin en soi, c'est un outil au service de l'efficacité de la création et de la manipulation du document de demain. C'est pourquoi la métaphore du bureau (elle-même proposée par A. Kay) va s'imposer, sur la base des éléments (fenêtres, dossiers, etc.) fournis par les travaux pionniers de l'ARC ; une métaphore explorée concrètement à l'aide de l'interface de la souris/pointeur.

Bref, on l'aura compris, la convivialité n'est pas native, elle émerge d'une torsion des outils au service d'une logique qui vise à faciliter le travail d'augmentation de l'intelligence (comme outil de gestion de la complexité) au bénéfice d'une logique qui travaille à réduire l'effort afin de rendre l'outil le plus « transparent »² possible pour qu'il soit utilisable par le plus grand nombre le plus aisément possible. Ce qui n'est jouable qu'en maximisant le potentiel intuitif et convivial des outils de base inventés par l'équipe de Douglas Engelbart (potentiel qui était maîtrisé par les contraintes du système). Une maximisation qui ne se fait plus au service du déplacement de l'effort afin de maximiser la gestion de la complexité,

1. BARDINI, *op. cit.*, p. 162.

2. Les couches d'interface graphique qui séparent l'utilisateur du code peuvent soit être vues comme un allègement parce qu'elles évitent à cet utilisateur d'investir dans la connaissance et la manipulation du code, soit comme une barrière puisqu'elles l'éloignent d'une « vraie » maîtrise de l'informatique qui ne peut passer que par celle du code.

mais au bénéfice d'une convivialité qui veut éviter au maximum l'effort et limiter au minimum le coût d'apprentissage.

NLS était un système de travail collaboratif. Il possédait donc des fonctions de communication et notamment l'une des premières fonctionnalités d'*e-mail*. Lorsque l'ARC accueille le Network Information Center d'Arpanet (en toute fin des années 1960), Douglas Engelbart envisageait que la communauté Arpanet devienne le prochain lieu de développement de NLS et de la logique du *bootstrapping*. Or, malgré cette situation carrefour, ce n'est pas la technologie NLS qui l'emporta, avec ses solutions toujours un peu plus sophistiquées/compliquées, mais l'*e-mail* mis au point par Ray Tomlinson à BBN en juillet 1970. . .

2. La modélisation, la résolution de problèmes et le réseau

On présente volontiers la pensée de J.C.R. Licklider comme largement orientée par la question de la communication. Elle est certes centrale¹, mais il ne faudrait pas qu'elle en vienne à masquer une autre dimension tout aussi fondamentale de sa réflexion, celle des modèles et de la modélisation. Une modélisation dont on retrouve de solides échos même dans les articles sur la communication et la symbiose. Mais surtout une modélisation dont la présentation sera beaucoup plus poussée dans ce qu'il appelle son « *pro-cognitive system* » (susceptible de constituer la bibliothèque du futur) et ses travaux des années 1970 sur une informatique modulaire. On a retenu surtout le réseau – lui-même plutôt dépendant de l'armée dans les années 1970, alors que Licklider le voyait irriguer l'ensemble de la société –, tout en oubliant que même la communication chez Licklider est d'abord une question de modélisation des représentations.

La modélisation ou la question des modèles² intervient en quelque sorte dans la vision de Licklider comme l'apprentissage chez Douglas Engel-

1. Je l'ai moi-même souligné dans Pascal ROBERT, « JCR Licklider et l'informatique de réseau(x) : imaginaire, impensé ou pensée visionnaire », dans : *Études de communication* 36 (2011), p. 111–128.

2. Car la question des modèles renvoie aussi bien à une modélisation plutôt « dure » (Cf. l'article de 1962 et le livre de 1965) qu'à une modélisation plutôt « molle » (de l'échange de modèles entre esprits).

bart. De même que l'apprentissage ne renie pas les logiques de facilitation mais empêche leur autonomisation dans une logique de pure convivialité, de même la modélisation ne renie pas celle de la communication (homme-machine et/ou réseautique) mais l'empêche de devenir une pure communication, un pur lien qui pourrait s'affranchir de la logique de la modélisation. Comme le souligne R. Fano,

He was interested in almost any aspect of human cognitive activities. While he became a knowledgeable and skillful programmer, his focus remained on the cognitive aspects of computer usage. He was fascinated by computers, because he understood in a deep way their potential to help people in a broad spectrum of activities by extending and amplifying their intellectual power, by improving the effectiveness of their communication and collaboration, and last but not least, by freeing them from the drudgery of clerical operations, to which he was particularly allergic.¹

C'est ainsi toujours la dimension cognitive, de résolution de problèmes et d'apprentissage qui l'a intéressé. Or, ces trois dimensions, à ses yeux, passent par la modélisation. La communication elle-même, dans un tel schéma ne peut être autonome, elle est inévitablement liée à la question des modèles qui, à son tour, ne doit pas plus prétendre à l'autonomie ou devenir une fin en soi, car l'une comme l'autre doivent être au service de la résolution de problèmes.

2.1 *On line man-computer communication*²

Ce qui focalise l'attention de Licklider c'est la *problem-solving*, la résolution de problème. Celle-ci tient en quelque sorte dans sa vision une place équivalente à celle de la complexité dans celle de Douglas Engelbart.

2.1.1 *Complementation et symbiose*

La résolution de problèmes passe chez Licklider par ce qu'il appelle une *complementation* entre l'ordinateur et l'homme. Nous sommes en

1. Robert M. Fano, *Joseph Carl Robnett Licklider 1915-1990, a biographical memoir*, Washington, DC: National Academy of Sciences, 1998, p. 20.

2. Titre de l'article de J.C.R. Licklider de 1962.

1962, c'est pratiquement nouveau et intellectuellement rarement revendiqué comme tel, sauf par Licklider lui-même dans son article de 1960 sur la symbiose. C'est pourquoi il écrit :

The fundamental aim in designing a man-computer symbiosis is to exploit the complementation that exists between human capabilities and present computer capabilities:

- (a) To select goals and criteria – human;
- (b) To formulate questions and hypotheses – human;
- (c) To select approaches – human;
- (d) To detect relevance – human;
- (e) To recognize patterns and objects – human;
- (f) To handle unforeseen and low probability exigencies – human;
- (g) To store large quantities of information – human and computer; with high precision – computer;
- (h) To retrieve information rapidly – human and computer; with high precision – computer;
- (i) To calculate rapidly and accurately – computer;
- (j) To build up progressively a repertoire of procedures without suffering loss due to interference or lack of use – computer.¹

Est-ce pour rassurer ? Mais une majorité des capacités listées relève de l'humain en propre (6 sur 10). Deux sont partagées par l'homme et la machine, à la précision près, ce qui signifie que cette dernière va l'emporter à terme. Enfin, deux renvoient exclusivement à l'ordinateur. Toute la question, que n'aborde pas Licklider, n'est-elle pas de savoir si l'ordinateur va remonter progressivement (puisqu'il parle de « *present computer capabilities* ») la liste et laisser la portion congrue à l'homme, dans une relation moins symbiotique que de domination ? Une domination qui ne serait pas seulement technique parce qu'elle aurait des conséquences directes sur la résolution de problèmes d'organisation socio-politique.

Cette symbiose va s'appliquer aux champs suivants : les mathématiques, l'informatique, les *war-* et *management gaming*, l'éducation² ; il ajoute

1. Joseph C. R. Licklider and Clark Welden, « On line man-computer communication », in: *Proceedings of the Spring Joint Computer Conference*, Cambridge, Mass.: AFIPS, 1962, URL: <http://doi.ieeecomputersociety.org/10.1109/AFIPS.1962.24>, p. 114.

2. Il ajoute également ce qu'il appelle les *semi-automatic laboratories*, sans expliquer de quoi il s'agit précisément.

que « *In the planning and design of systems of many kinds, digital simulation is recognized as a valuable technique* »¹. Bref, un univers de résolution de problèmes, pratiques et conceptuels qui passe le plus souvent par des opérations de modélisation.

2.1.2 Trois applications

L'article présente ensuite trois applications dans lesquelles l'articulation entre homme et machine à la fois exige et permet d'améliorer la modélisation des activités.

La modélisation des échanges de communication dans l'enseignement. Modéliser la relation homme/machine permet d'élaborer des systèmes d'aide à l'enseignement des langues et de l'apprentissage des mathématiques notamment.

La simulation hospitalière, le co-planner. Ce qu'il appelle un co-planner est un dispositif qui articule la technique (la machine), les données et les programmes qui permettent de les « mouliner ».

Encore une fois Licklider fait bien attention à éviter la critique d'une machine qui remplacerait l'homme : si elle simule, c'est pour aider la prise de décision et non s'y substituer. C'est pourquoi « *the computer, parts of the system are not intended, we should emphasize, to calculate optimal plans or designs ; they are intended to provide memory, manipulative, computing, and display functions in such a way that they can be integrated with the more intuitive functions supplied by the human parts of the system.* »²

Visualizing the operation of computer programs. Enfin, Licklider envisage d'appliquer la démarche de résolution de problèmes/modélisation à l'informatique elle-même : « *the conclusion, therefore, is that it might be interesting to experiment with programs that display various aspects of the internal operation of the running computer* »³. Cette boucle réursive, l'auteur la compare volontiers à une « introspection », mais ce retour n'est-il pas moins une conscience qu'un simple outil de contrôle ?

1. *ibid.*, p. 114.

2. *ibid.*, p. 118.

3. *ibid.*, p. 119.

En conclusion, il revient, mais cette fois sous la forme d'une métaphore (celle de l'œuf et de la poule) sur son idée d'une symbiose. En effet, écrit-il,

[...] it appears that the development of effective human cooperation and the development of man-computer symbiosis are "chicken-and-egg" problems. It will take unusual human teamwork to set up a truly workable man-computer partnership, and it will take man-computer partnerships to engender and facilitate the human cooperation. For that reason, the main tasks of the first time-sharing computer system with many remote stations may well be in the areas of language and procedure development.¹

S'il s'agit, dans cet article, d'améliorer la communication homme/machine (H/M) réseautique ce n'est pas contre ou au détriment de la démarche rationnelle/scientifique. Au contraire, Licklider veut améliorer la capacité de résolution de problème de l'homme en société. Autrement dit, améliorer l'interface H/M c'est améliorer cette aptitude en croisant les capacités de l'homme et de la machine et en augmentant la vitesse d'exécution des opérations. C'est aussi rendre dynamique les opérations de simulation. Bref, il entrecroise communication et modélisation, en ce sens que la communication (facilité d'utilisation et ce, à distance) est au service de la modélisation comme la modélisation (dynamique) est au service de la communication, tout en gardant à l'esprit que les deux sont au service de la résolution de problème et d'aide à la décision.

2.2 Le *procognitive system*

Au début des années 1960 (de 1960 à 1962) J.C.R. Licklider, alors chez BBN (Bolt Beranek et Newman²), est chargé d'une étude commanditée par le Council on Library Resources (une organisation créée par la Ford Foundation en 1956). Cette étude vise à définir ce que sera la bibliothèque du futur, celle des années 2000, dit-il³. Il souligne que la science des bibliothèques de l'époque n'offre ni concepts ni système qui serait efficace

1. *ibid.*, p. 119.

2. Une des sociétés les plus dynamiques en matière d'innovation informatique et de réseaux des années 1960.

3. Joseph C. R. LICKLIDER, *Libraries of the Future*, Cambridge, Mass. : MIT Press, 1965.

ou *desirable* (souhaitable en français ou moins littéralement « que l'on aurait envie de promouvoir ») : constat plutôt négatif. Ce sera l'occasion de revenir sur les outils du savoir et de l'accumulation du savoir développés jusque-là – notamment sur la bibliothèque et le livre comme outils de modélisation/présentation/diffusion de la connaissance. Il présente un modèle simplifié de leurs propriétés essentielles, dont assez paradoxalement il souligne les limites (ce qui les disqualifie) tout en conservant une large part des aptitudes acquises de modélisation des connaissances. Il en vient, bien évidemment, à présenter un système informatisé qu'il nomme un *pro-cognitive system* pour en souligner les qualités dynamiques.

2.2.1 Page, livre et bibliothèque

Ce qui intéresse Licklider ce n'est ni le fait que l'information ou la connaissance soit imprimées sur papier, ni même les mots et les phrases, mais « *the facts, concepts, principles, and ideas that lie behind the visible and tangible aspects of documents* »¹. Ce qu'il appelle une *transformable information*, c'est-à-dire une information que l'on peut transporter d'un support à l'autre sans déperdition, ou avec une déperdition/dégradation minimale. Tout ce qui relève de l'art s'y prête particulièrement mal, car même une reproduction est une transformation. Même la littérature y perd beaucoup, ce n'est véritablement valable que pour les activités scientifiques, techniques, l'histoire, la médecine, le droit et la gestion (des entreprises et de l'État).

Selon lui, les hommes pensent en suivant des schèmes (*schemata*), qui s'articulent en trois niveaux : le système, les sous-systèmes et les composants. L'innovation doit se faire en réutilisant et modifiant le niveau des composants. Or, dans le domaine de la bibliothèque : le système c'est la bibliothèque, les sous-systèmes, les livres et les composants, les pages.

D'ailleurs dans ces trois niveaux Licklider considère que la page est un bon outil d'affichage (« *as a medium for display of information, the printed page is superb* »²), alors que le livre est volumineux (*bulky*) et lourd (*heavy*) ; bref, « *books are not very good display devices. In fulfilling the storage function, they are only fair. With respect to retrievability they*

1. *ibid.*, p. 2.

2. *ibid.*, p. 4.

are poor »¹. La bibliothèque, qui regroupe des livres, ne peut être un outil satisfaisant : « *if books are intrinsically less than satisfactory for the storage, organization, retrieval, and display of information, then libraries of books are bound to be less than satisfactory also.* »²

Livres et bibliothèques sont d'autant moins performants qu'ils ne correspondent pas du tout à ce qu'est, aux yeux de Licklider, l'interaction avec ce qu'il appelle le *body of knowledge* : « *conceived of as a dynamic process involving repeated examinations and intercomparisons of very small and scattered [dispersées] parts* ». Cette vision est singulièrement proche de la logique du fragment qui domine aujourd'hui sur Internet. Il reproche même à la page imprimée sa passivité (ce qui est oublier la dynamique de l'index et de la note de bas de page, voire de l'encadré).

Il en va selon lui de deux difficultés : celle de séparer l'information de la page elle-même et l'absence de ce qu'il appelle des *active processors*. C'est pourquoi il veut substituer (il emploie le verbe *to substitute*) au livre un dispositif qui transmette/diffuse l'information sans transport matériel et qui, non seulement présente l'information aux gens, mais la traite aussi pour eux. Ce substitut se composera d'une conjugaison de bibliothèque et d'ordinateur. Il qualifie un tel dispositif de *procognitive system*³, et précise en note : « *the system in which we are interested are broader than present day libraries ; the system will extend farther into the process of generating, organizing and using knowledge.* »

Il rejette la page comme outil de stockage de l'information à long terme mais pas à court terme. D'ailleurs il conserve un certain nombre de *schèmes* hérités de la page imprimée :

- la hiérarchie de segments de textes (caractère, mot, paragraphe, chapitre etc.) ;
- les concepts d'information textuelle, tabulaire, graphique et picturale/imagière ;
- des concepts comme ceux d'auteurs, d'abstracts, notes de bas de page et listes de références ;

1. *ibid.*, p. 5.

2. *ibid.*, p. 5.

3. *ibid.*, p. 5.

- des concepts comme article original, compte rendu, note, lettre, journal et livre (« *in the sense of classes of information, not physical carriers of information* ») ;
- des concepts comme catalogue, index, descripteur et thésaurus.

Paradoxalement, au final, il ne rejette pas grand-chose du monde ancien du livre et des bibliothèques, si ce n'est leur concrétisation elle-même... mais n'est-ce pas parce qu'il veut introduire un nouveau support, l'ordinateur ? Nous allons voir cependant que les choses sont un peu plus subtiles.

2.2.2 Pertinence de l'informatique

S'il récusé (partiellement) le vieux monde de la bibliothèque et du livre, il rejette également une conception déjà dépassée à ses yeux de l'informatique. Il ne veut plus d'une informatique de données, de programmes, de cartes perforées, que l'on traite dans un centre informatique et qui ne peut être l'outil de concrétisation de son *procognitive system*. Il lui faut une autre informatique qui repose sur les éléments suivants :

- Random acces memory
- Content adressable memory
- Parallel processing
- CRT displays and light pens
- Procedures, subroutines and related components of computer programs
- Hierarchical and recursive program structures
- List structures
- Procedure-oriented and problem-oriented languages
- Xerographics output units
- Time-sharing computer systems with remote stations

La philosophie de sa démarche réside toute entière dans cette idée : « *what is of value for our purpose is not, for example, the oscilloscope or the light pen. It is the schema in which a man sits at a desk, writes or draws on a surface with a stylus, and thereby communicates to a programmed information processor with a large memory* ». Ce n'est, pas plus qu'avec la bibliothèque et le livre, la concrétisation qui compte, c'est le concept, le schéma fonctionnel du dispositif, qui est fondamentalement un dispositif d'écriture-lecture et de communication (questions-réponses immédiates)

avec l'ordinateur, que ce soit à travers les techniques de l'époque, dont Licklider connaît à la fois le potentiel et les limites...ou celles de demain (d'aujourd'hui par exemple)! C'est pourquoi il souligne qu'il est « *important to recognize that our progress, must for a time, be largely conceptual* ».

2.2.3 The procognitive system

J.C.R. Licklider part du principe que c'est le système entre l'ordinateur, l'homme et ce qu'il appelle le *body of knowledge* qui doit être pris en compte. Pas l'une ou deux seulement des trois composantes. On ne doit pas non plus reconduire l'approche traditionnelle qui sépare certaines fonctions allouées à l'homme et d'autres dévolues à la machine. Il faut tisser les deux ensemble nous dit Licklider (c'est lui qui emploie la métaphore de la couverture¹). Il pousse ainsi un peu plus loin encore le programme de l'article de 1962 qui raisonnait, rappelons-le, en termes de « compensation ».

Son *procognitive system* requiert un médium physique, des langages, et une capacité d'adaptation, voire d'auto-organisation. Je vais me focaliser sur le premier point. Licklider préfère le terme *intermedium* à celui d'interface qui, selon lui, renvoie à la notion de surface de séparation justement. *Intermedium* est censé dire le lien, la relation (en écho avec sa philosophie de la symbiose), et être la base d'un véritable *personal documentation system*.

Les prérequis du système sont les suivants :

- Un moniteur/écran couleur ou NB (« *each element of the display should be selectively erasable by the computer program, and also either directly by the operator* »).
- Il doit permettre d'effectuer des « *hard copy* », des copies permanentes.
- Il doit permettre de faire des dessins et des graphiques, grâce à un stylet (« *if the stylus is connected by a wire to console, the wire should be very light and flexible and should not constrain the manipulation of the stylus* » – ce qui nous semble une *évidence* aujourd'hui et ne l'est pas à l'époque),

1. *ibid.*, p. 91.

- Le dispositif doit également être fiable et économiquement viable.
- Il doit aussi posséder une interface d'écriture, un clavier ; Licklider souligne que, dès lors qu'un clavier est connecté à un ordinateur, il n'y a plus de raison pour que ce qui est tapé soit immédiatement imprimé sur une feuille de papier ; il peut l'être, mais va d'abord être affiché sur l'écran – là encore, ce qui nous paraît une *évidence* aujourd'hui mérite une remarque précise de l'auteur à l'époque.

Licklider prévoit que son *procognitive system* ne doit pas faire l'objet que d'une seule utilisation individuelle. C'est aussi un outil de travail en groupe, un outil de travail collaboratif (pensons à NLS d'Englebart). Il envisage deux solutions :

- Des consoles reliées entre elles selon des techniques qu'il renvoie explicitement au téléphone et à la TV (alors que ses idées par ailleurs mèneront au développement d'une logique de réseau purement informatique).
- Un dispositif de projection sur grand écran, avec un dérivé du *light pen* pour la communication entre les membres du groupe et la machine et un puissant système de zoom (Power-point et Google maps tout ensemble avant la lettre ?).

2.3 La fin des années 1960

Qu'est-ce donc que « communiquer » pour notre auteur dans son article de 1968 « Computer as a Communication Device » ? Communiquer, c'est échanger des modèles informationnels à propos de quelque chose. Des modèles qui doivent être sanctionnés par la société. Autrement dit, la communication est définie par Licklider comme une opération de *cooperative modeling*. Opération soutenue par des moyens d'*externalization* : « *Even such a simple externalized model as a flow diagram or an outline – because it can be seen by all the communicators – serves as a focus for discussion. It changes the nature of communication.* »¹

L'ordinateur ne doit pas être utilisé seulement comme outil d'automatisation de tâches. C'est ce à quoi aboutit Licklider, après avoir pris connais-

1. Joseph C. R. LICKLIDER et Robert W. TAYLOR, « The computer as a communication device », dans : *Science and Technology* (1968), p. 20–41.

sance des travaux de D. Engelbart sur les interfaces et l'interaction (le système souris-pointeur par exemple).

We can say with genuine and strong conviction that a particular form of digital computer organization, with its programs and its data, constitutes the dynamic, moldable medium that can revolutionize the art of modeling, and that in so doing can improve the effectiveness of communication among people so much as perhaps to revolutionize that also.¹

Soulignons que l'ordinateur est vu, dès cette époque comme un *medium* par Licklider, ce qui est pour le moins nouveau. Et ce médium est à la fois modélisé/modélisant et pensé comme ce qui va « révolutionner » l'exercice même de la modélisation. Or, cette modélisation ne s'oppose pas à la communication qui, pour le psychologue qu'est avant toute chose JCR. Licklider est échange de modèles de représentations. C'est pourquoi il revient encore une fois sur l'idée que « *creative, interactive communication requires a plastic or modable medium that can be modelled, a dynamic medium in which premises will flow into consequences, and above all a common medium that can be contributed to and experimented with by all[...]* »²

Toujours en cette fin des années 1960, Licklider avait à l'esprit un ordinateur qui pourrait être utilisé par des chercheurs non spécialisés en informatique, avec un minimum de programmation. Or, ce qui intéresse un chercheur d'une autre discipline selon Licklider, c'est de pouvoir utiliser l'ordinateur comme un outil de modélisation. Un outil de modélisation avec lequel on peut interagir directement grâce à un dispositif d'interface graphique. Le chercheur aurait à sa disposition (et/ou créerait) une bibliothèque de modules logiciels qu'il pourrait ensuite utiliser comme des outils au service de ses besoins de modélisation. Chaque module serait indexé pour pouvoir être facilement réutilisé. Vision puissante, mais fort difficile à mettre en œuvre à l'époque. Licklider s'est beaucoup investi personnellement dans cette démarche entre 1969 et 1974 sur un financement IPTO, et sa bibliothèque a atteint les 2 000 modules... Cependant, l'ARPA avait pour objectif de financer des projets dans leur phase de lancement,

1. *ibid.*, p. 27.

2. *ibid.*, p. 22.

c'est-à-dire cinq années. Dès lors, après 1974, Licklider se retrouva sans financement... et le projet périclita ; d'ailleurs il n'écrivit aucun article ni rapport sur ce projet... on peut y voir plus qu'une déception.

Aujourd'hui, la prégnance d'Internet – et plus globalement celle des réseaux (d'informatique et de télécommunications) – est telle que l'on relit volontiers le travail de Licklider à l'aune de la question de la communication en oubliant peut être que la pensée de Licklider s'adressait autant aux questions de la résolution de problèmes et de la modélisation. La communication homme/machine et réseautique n'étant, d'une certaine manière, qu'un outil au service de leur efficacité. Certes, Licklider a publié un article sur les réseaux¹ en 1963 et il a été le directeur de l'IPTO... mais pendant deux ans seulement (de 1962 à 1964) bien avant le lancement d'ARPANET et alors que son grand projet – MAC – est un projet de *time sharing* ; certes, ses idées, reprises par Robert Taylor et Larry Roberts, ont participé au développement d'ARPANET, mais les réseaux qu'envisage Licklider relèvent largement d'une technologie, le *time-sharing*, qu'il promeut dans ses travaux de 1960/1962 et 1968, et qui sera amplement dépassée dès la fin des années 1960. ARPANET, en ce sens, n'est pas une application des idées de Licklider. C'est pourquoi il n'est pas aberrant de considérer que le poids rétrospectif que font peser les réseaux en vient à opérer une *torsion* non négligeable de sa vision de l'ordinateur conçu en premier comme outil de résolution de problèmes/modélisation. Une torsion qui oublie que cette logique met la communication à son service (au moins autant que l'inverse) et qu'elle leste le projet de Licklider d'un poids (celui d'investissements lourds) et d'un sérieux (dans le management des projets) qui participent certes à alléger les apprentissages mais, là encore pour aller plus loin et plus vite, c'est-à-dire sans les renier.

Conclusion

En définitive, l'histoire de l'informatique des années 1970 correspond à une torsion des visions de D. Engelbart et JCR. Licklider (on peut aussi

1. Joseph C. R. LICKLIDER, *Memorandum for Members and affiliates of the Intergalactic Computer Network*, Advanced Projects Research Agency, 1963, URL : <http://www.kurzweilai.net/memorandum-for-members-and-affiliates-of-the-intergalactic-computer-network>.

considérer que la déformation de ces visions s'offre comme un outil de lecture/décryptage de cette histoire) qui aboutit à la mise entre parenthèse de l'apprentissage et de la modélisation au bénéfice de la convivialité ou de la communication dans le cadre de systèmes plutôt fermés (d'abord chez Xerox puis avec Apple). On peut alors se demander si le Libre ne correspond pas à la « reprise » (mais sans forcément assumer une continuité) de la logique de l'apprentissage et de la modélisation (au sens large). Car s'il est « libre » c'est aussi sur le fond de l'exigence d'un accès difficile qui nécessite un long et intense apprentissage et d'un fort investissement technique (quand bien même certains l'oublient parce qu'ils ne le vivent pas comme tel). Autrement dit, la « liberté » technico-sociale appelée de ses vœux par le mouvement du Libre est en quelque sorte payée par le prix de l'acquisition d'une compétence technique élevée. Ne butera-t-elle pas toujours sur une convivialité et une communication peut-être aliénantes, mais socialement et cognitivement moins coûteuses ?

Références

- BARDINI, Thierry, *Bootstrapping : Douglas Engelbart, Coevolution, and the Origins of Personal Computing*, Stanford : Stanford University Press, 2001.
- ENGELBART, Douglas, *Augmenting human intellect : a conceptual framework*, Stanford : SRI, 1962, URL : <http://www.dougenelbart.org/pubs/augment-3906.html>.
- ENGELBART, Douglas et William ENGLISH, « A research center for augmenting human intellect », dans : *The New Media Reader*, sous la dir. de Noah WARDRIP-FRUIIN et Nick MONTFORT, Cambridge Mass. : MIT Press, 2003, p. 231–246.
- FANO, Robert M., *Joseph Carl Robnett Licklider 1915-1990, a biographical memoir*, Washington , DC : National Academy of Sciences, 1998.
- FOUCAULT, Michel, *L'archéologie du savoir*, Paris : Gallimard, 1969.
- KAY, Alan et Adele GOLDBERG, « Personal dynamic media », dans : *The New Media Reader*, sous la dir. de Noah WARDRIP-FRUIIN et Nick MONTFORT, Cambridge Mass. : MIT Press, 2003, p. 391–404.
- LICKLIDER, Joseph C. R., *Libraries of the Future*, Cambridge, Mass. : MIT Press, 1965.

LICKLIDER, Joseph C. R., « Man-computer symbiosis », dans : *IRE Transactions on human Factors in Electronics* 1 (1960), p. 4–11.

– *Memorandum for Members and affiliates of the Intergalactic Computer Network*, Advanced Projects Research Agency, 1963, URL : <http://www.kurzweilai.net/memorandum-for-members-and-affiliates-of-the-intergalactic-computer-network>.

LICKLIDER, Joseph C. R. et Robert W. TAYLOR, « The computer as a communication device », dans : *Science and Technology* (1968), p. 20–41.

LICKLIDER, Joseph C. R. et Clark WELDEN, « On line man-computer communication », dans : *Proceedings of the Spring Joint Computer Conference*, Cambridge, Mass. : AFIPS, 1962, URL : <http://doi.ieeecomputersociety.org/10.1109/AFIPS.1962.24>.

ROBERT, Pascal, « JCR Licklider et l'informatique de réseau(x) : imaginaire, impensé ou pensée visionnaire », dans : *Études de communication* 36 (2011), p. 111–128.

– *Mnémotechnologies, pour une théorie générale critique des technologies intellectuelles*, Paris : Hermès, 2010.

Sharing Source Code

Christopher KELTY, *Two Bits, The Cultural Significance of Free Software*, chapitre « Sharing Source Code », Duke University Press, 2008, pp. 118-143. URL : <http://twobits.net>. Traduction par Camille PALOQUE-BERGES

Ce chapitre étant placé sous licence Creative Commons By-NC-SA, vous pouvez télécharger gratuitement sa traduction sur <http://framabook.org>.

PRÉSENTATION

Par Camille PALOQUE-BERGES

Si l'histoire des technologies logicielles est bien documentée, celle du logiciel libre l'est beaucoup moins, explique dans ce chapitre Christopher Kelty en s'attaquant à la saga d'Unix.

Une des raisons en est le fait que la « philosophie » qui lui est attachée est tissée de narrations et de représentations, un imaginaire hybride à la croisée de la technique et de l'idéologie – et, prenant cette voie, le propos de Kelty s'inscrit dans la trajectoire de la sociologie culturelle appliquée à l'imaginaire des technologies informatiques.

Ce texte est au cœur de l'ouvrage *Two-Bits, The Cultural Significance of Free Software* publié en 2008 (sous licence Creative Commons-By-NC-Sa) ; il décrit le deuxième moment d'une partie centrale détaillant les cinq principes qui, selon Kelty, sont les piliers explicatifs de cette philosophie : le « mouvement », émergeant à partir d'un certain nombre de valeurs et de pratiques liées au monde des technologies informatiques de la fin des années 1990 (de l'*open source* aux navigateurs dits « libres ») ; le « partage du code source », un retour en arrière sur les premières et multiples formes de la circulation des codes informatiques des années 1960 aux années 1980, prenant comme paradigme le système d'exploitation UNIX (objet du texte traduit ci-dessous) ; la « conception des systèmes ouverts » revenant sur la manière dont la notion d'ouverture a été inventée et conceptualisée ; l'apparition et la systématisation d'une « écriture propre aux licences du libre » ; et enfin, les modes de « coordination des formes de la collaboration ».

Kelty procède par une analyse culturelle à la croisée des discours ayant accompagné et conditionné dans une très large mesure les exploits informatiques des adeptes du logiciel libre et des réalisations techniques à proprement parler. C'est pourquoi les « geeks » qui peuplent l'ouvrage sont moins une figure folklorique que les moteurs d'un développement à la fois social et technique d'une série d'alternatives aux trajectoires d'innovation et de développement des technologies informatiques académiques et commerciales ; sans territoire ni milieu propres, mais cependant marqués par des effets de structure et de pouvoir, ils sont partout comme le code source qu'ils partagent, et sont à la base de la notion de « public récursif », concept central de l'ouvrage.

Selon Kelty, un public récursif se constitue autour d'un besoin commun de maintenir les moyens d'association qui lui permettent de se penser comme public. Chez ces publics, la technologie est utilisée comme un argument, un discours, aussi bien qu'elle l'est pour des raisons instrumentales, comme le précise Kelty : le public parle de technologie, mais exprime aussi des idées grâce à elle. Ce public conçoit également les infrastructures au travers desquelles les idées peuvent circuler et être exprimées. Internet et le logiciel libre sont ainsi deux environnements privilé-

giés de ces publics, à la fois développés, regardés, commentés, et habités par eux.

Remontée généalogique aux origines de ces publics, ce chapitre a aussi une valeur archéologique dans la mesure où il va fouiller dans les couches profondes du code pour mieux comprendre les pratiques de la programmation, ainsi que le savoir-faire technique et les croyances technologiques qui les sous-tendent.

II

Économie du Libre

Les modèles économiques du logiciel libre et leur évolution

Stéphane RIBAS,
Patrick GUILLAUD,
Stéphane UBÉDA

*Un logiciel libre est gratuit une fois
qu'il a été payé*

François Élie

Le mouvement du logiciel libre représente actuellement environ six cent mille projets et plus de cent millions de développeurs. Plus de la moitié des solutions et des infrastructures installées par les entreprises dans les cinq ans à venir seront basées sur du logiciel libre. 2011 a été une année record pour les investissements dans ce domaine avec une augmentation de 49% pour atteindre 675 millions de dollars. Les nouvelles entreprises de logiciel s'appuient de plus en plus sur le logiciel libre qui leur permet de générer du chiffre d'affaires avec des modèles économiques adaptés.

Ces chiffres et projections issus d'études récentes réalisées par Blackduck Software (*6th Annual Future of Open Source Survey*) donnent une idée de l'ampleur du mouvement du libre qui est aujourd'hui bien installé

dans le paysage économique. Ce chapitre apporte des réponses à la désormais classique question de savoir comment générer du chiffre d'affaires et des revenus à partir de logiciels qui sont ouverts et gratuits.

À l'instar du réseau routier dont la fonction première est de faciliter les déplacements des usagers indépendamment de leurs activités, les logiciels libres sont indépendants des modèles économiques liés à leur utilisation. Une conséquence importante est que le mouvement du logiciel libre offre deux types de retour sur investissement : le premier répond au modèle de l'édition logicielle (amortissement des coûts de développement) et l'autre, moins facile à quantifier, intervient à travers l'impact sociétal produit, qui peut être parfois considérable (par exemple Internet).

Dans ce chapitre nous présenterons, en adoptant une perspective historique, le concept d'*openness* et les différents modèles économiques associés au logiciel libre : services, mutualisation, valeur ajoutée, double licence, fondation. Seront également évoqués des modèles basés sur le matériel, le *cloud*, l'assemblage de composants et les terminaux mobiles. Avant d'aborder ces questions nous rappellerons quelques concepts clés.

Qu'est-ce qu'un modèle économique ? Pour en donner une définition très succincte, un modèle économique est une description des mécanismes de génération de revenus, que ce soit pour des personnes physiques (artisans, etc.) ou morales (entreprises, collectivités, etc.). Les modèles économiques sont aussi vieux que la société : le marchand itinérant qui augmente la valeur perçue de ses produits en les déplaçant d'un point à un autre, la place du marché où les commerçants produisent mutuellement de la valeur en concentrant des produits, donc du choix, dans le temps et dans l'espace, l'échoppe de quartier qui offre un point d'approvisionnement stable pour les produits courants, le travail à façon qui mobilise les compétences spécifiques de l'artisan, etc. Les modèles économiques rencontrés dans le monde du logiciel libre sont liés à une gestion de la propriété intellectuelle spécifique puisque le principe des licences du logiciel libre consiste en un abandon contrôlé, parfois imposé, des droits patrimoniaux des auteurs.

La non rivalité et l'*openness*. Le logiciel libre est fondé sur deux principes essentiels : la « non rivalité » et l'*openness*. Un bien est dit « non

rival » lorsqu'il peut être transmis sans dépossession, comme la connaissance. Les logiciels appartiennent aussi à cette catégorie : on peut copier et transmettre un logiciel autant de fois que l'on veut tout en continuant à s'en servir soi-même. Le concept d'ouverture ou *openness* peut être défini comme l'attitude consistant à systématiser la diffusion et le partage des biens non rivaux. Ce type de comportement, dont l'exemple le plus emblématique est celui de la recherche scientifique, accélère considérablement la production et la diffusion des connaissances et augmente significativement l'innovation. La réponse que fournit l'*openness* à la question du mode de financement s'appelle *écosystème* et nous verrons que les implications de cette réponse excèdent largement le seul financement du développement du logiciel.

Le prix, le coût et la valeur perçue. Tout modèle économique fait intervenir au moins trois grandeurs fondamentales. Le prix d'un produit ou d'un service qui est déterminé par un vendeur en fonction d'un contexte de marché, c'est le prix « affiché sur l'étiquette ». Le coût d'un produit ou service qui correspond à la somme des ressources nécessaires à celui qui le fabrique ou l'élabore pour le produire ou le fournir. La valeur perçue d'un produit ou service est l'évaluation par un acteur donné de l'intérêt que présente pour lui l'acquisition de ce produit ou ce service. Le prix résulte donc d'une décision, le coût d'une analyse et la valeur perçue d'une évaluation. Ces trois variables sont relatives, elles dépendent de chaque acteur et déterminent des indicateurs (ex. marge ou perte) et des comportements (ex. décision d'achat ou non). La nature des articulations entre ces trois grandeurs et leur organisation dans le temps et dans l'espace sont caractéristiques d'un modèle économique.

Réseau routier, Internet et logiciel libre. La fonction d'un réseau routier est de faciliter la circulation, quel que soit le type des véhicules qui l'empruntent (vélo, voiture, bus, camion, etc.), et indépendamment de la nature de l'activité (trajet privé pour se rendre à l'école ou au travail, trajet professionnel, transport en commun, de fret, etc.). Partant de ce constat on peut imaginer de modéliser le réseau routier, les véhicules qui l'empruntent et la nature des déplacements, en une série de « classes » d'éléments. Le réseau routier est une instance d'une classe « infrastructures »

contenant toutes sortes de réseaux : le réseau Internet, d'électricité, réseau ferré, etc. De la même façon on peut considérer une classe « types de véhicules routiers » contenant une collection de dispositifs adaptés à la circulation sur l'instance « réseau routier ». Une classe « applications » contiendrait des instances de type « service de transport de personnes », « déplacement individuel privé », etc. On peut envisager une classe « types de protocoles » de l'infrastructure « Internet » qui correspondrait à la classe « type de véhicules » de l'infrastructure « réseau routier ». Il est également possible de dessiner une séparation entre les classes de niveau inférieur qui concernent l'infrastructure et celles de niveau supérieur concernant son utilisation. L'extension de cette analogie au domaine du logiciel permet de constater que le logiciel libre se situe souvent au niveau structurel : il est le composant essentiel de l'infrastructure « Internet » (dont la description de ses protocoles, les « véhicules » qui y circulent) et concerne la majeure partie des briques de base des systèmes d'information constituant le web (GNU/Linux, Apache, MySQL, PHP, etc.).

Il apparaît que les acteurs qui contribuent aux infrastructures et ceux qui contribuent aux applications ne sont pas les mêmes. Les premiers sont généralement des institutions de recherche, des universités dont l'objectif initial n'est pas de réaliser des profits financiers immédiats mais de faire progresser les connaissances de manière collaborative, ouverte et mutualisée en créant des infrastructures à disposition de la société tout entière et sur lesquelles le secteur privé peut s'appuyer notamment pour exercer des activités commerciales. Cette complémentarité des modèles, à laquelle nous devons notamment Internet et le Web, est d'une remarquable efficacité puisque l'on estime aujourd'hui à 3 500 milliards de dollars le chiffre d'affaires généré notamment par les 113 millions de sites web actifs sur l'infrastructure Internet¹.

L'évocation de ces chiffres astronomiques est l'occasion de remarquer qu'une approche financière directe est extrêmement réductrice car elle est incapable de mesurer d'autres effets comme les interactions entre personnes ou la formation et la diffusion des connaissances, ou encore la circulation de l'information, qui sont pourtant des facteurs clés pour l'innovation. Le parcours historique qui suit va nous permettre de présenter

1. Eben MOGLEN, *Innovation under Austerity*, F2C Freedom to Connect, 2012.

les moyens qu’ont trouvés les acteurs du logiciel libre pour générer de la valeur.

1. Années 1960 et 1970, les débuts de l’informatique : des modèles économiques basés sur le matériel

Durant les décennies de 1960 et 1970, le marché percevait la valeur de l’informatique comme étant concentrée dans le matériel¹. Les logiciels constituaient un complément nécessaire permettant de rendre les ordinateurs utilisables. Les systèmes d’exploitation, destinés à animer les machines, et les utilitaires, étaient considérés comme secondaires : leur finalité était de permettre aux constructeurs de valoriser leur offre matérielle. Les logiciels alors financés par les constructeurs d’ordinateurs étaient fournis avec les machines. De ce fait, les programmeurs étaient libres de mutualiser et d’échanger leurs codes. Ce fut le temps des premiers hackers et le développement logiciel s’inspirait du modèle de la recherche basé sur l’ouverture, le partage et la réutilisation. Les questions liées à l’appartenance, à l’encadrement de leur utilisation, à leur modification ou leur diffusion n’étaient pas à l’ordre du jour. Cette situation a perduré tant que les revenus des constructeurs d’ordinateurs se sont limités à la vente d’équipements matériels spécifiques. Cependant, avec l’évolution et la complexification des machines, les coûts de développement augmentèrent et les programmes livrés par les constructeurs dans des versions minimales ne satisfirent plus tous les utilisateurs. Les grands laboratoires de recherche et les universités commencèrent alors à développer des logiciels pour mieux répondre à leurs besoins, enrichissant ainsi l’offre des constructeurs de codes plus complets, plus performants et qui circulaient librement. Un exemple emblématique est le système d’exploitation Unix, développé par les laboratoires Bell à la fin des années soixante et commercialisé à partir de 1975. Caractérisé par sa portabilité – il ne dépendait pas d’un constructeur – et par sa structure modulaire, Unix se diffusa rapidement dans le monde universitaire. L’université de Californie à Berkeley qui disposait des codes sources fit évoluer le système en adoptant un

1. Nous indiquerons en note de bas de page le modèle économique mis en œuvre, précédé de « ME ». Dans ce cas il s’agit de la vente d’un ensemble indissociable constitué d’un équipement matériel et des logiciels permettant de l’exploiter.

modèle collaboratif de développement qui préfigurait l'organisation des communautés *open source*. Cette initiative fut à l'origine de la première distribution Unix, la Berkeley Software Distribution ou BSD. L'association de BSD et des ordinateurs VAX de Digital Equipment, souvent mis à disposition des laboratoires universitaires par ce constructeur, en ont fait le premier système informatique de référence dans le milieu académique. Un détail d'importance pour la suite fut l'adjonction par les développeurs de Berkeley du support de TCP/IP dans BSD, faisant ainsi de chaque ordinateur VAX installé une machine « *Internet ready* » près d'une décennie avant l'adoption d'Internet par le grand public. Un autre point à noter est la grande attention portée à la qualité du manuel d'utilisation d'Unix dont le format standardisé respectait des règles extrêmement strictes. Ce soin porté à la rédaction du manuel est souvent cité comme l'un des facteurs clés de succès d'Unix.

À cette époque Richard Stallman, chercheur au MIT et spécialiste de la programmation, des langages et des systèmes d'exploitation, commença à réfléchir à la création d'une licence garantissant le libre usage des logiciels, la libre accessibilité de leur code et réservant aux utilisateurs la possibilité d'étudier, de modifier, d'améliorer et de redistribuer les codes modifiés. Stallman annonça également son ambition de créer un système Unix dont tous les composants seraient libres. Plusieurs de ses collègues soutinrent cette initiative et en 1976 le groupe diffusa le premier éditeur libre Emacs, puis un an plus tard le gestionnaire de dépendances Make, suivi d'un débogueur puis d'autres outils. Le groupe fut rejoint par un nombre croissant de programmeurs à travers le monde¹. La dissémination d'Unix et la standardisation de l'informatique professionnelle provoquèrent un foisonnement de petites entreprises spécialisées dans le développement d'applications spécifiques et d'édition de logiciels standards. Cela constitua le point de départ du développement d'un nouveau secteur économique d'autant plus prometteur que l'évolution rapide de la taille du parc de machines laissait entrevoir un marché potentiellement gigantesque.

En somme, les modèles économiques des décennies 1960 et 1970 (et un peu au-delà) se sont construits autour du système Unix. C'était le règne du matériel et la période des architectures client/serveur. La source principale

1. ME : création d'un écosystème.

de revenus des constructeurs informatiques était la vente d'équipements matériels, le logiciel étant « offert »...

2. Années 1980, l'époque des pionniers du Libre

Au début des années 1980 le système d'exploitation Unix de Bell Labs dominait largement le marché américain. On dénombrait 400 installations de systèmes Unix dans les universités en 1982 (contre 3 en 1979) et ce chiffre n'a cessé d'augmenter pendant les décennies 1980 et 1990. La plupart des étudiants qui travaillaient et développaient sur ce type de système utilisaient les outils « libres » avant de partir travailler en entreprise où ils trouvaient – ou mettaient en place – les mêmes environnements¹. La licence du système Unix, qui accorde le droit d'utiliser le système, était achetée à Bell à un prix modeste mais, une fois cette formalité remplie, la plupart des utilisateurs installait BSD, la distribution universitaire du système qui rassemblait tous les utilitaires libres. La diffusion de ces logiciels empruntait un nouveau vecteur : Usenet, un réseau communautaire d'échange de messages, de programmes, de support, entre des centaines de programmeurs. Le déploiement d'Usenet, favorisé par le support de TCP/IP par la plupart des machines sous Unix, généralisa la diffusion des programmes par voie numérique et en même temps installa l'idée qu'un logiciel est naturellement et légitimement libre. L'architecture de Usenet était non centralisée, le réseau diffusant les contenus de proche en proche sur les réseaux des universités mais également entre universités. Ce furent les prémices d'Internet. En 1980 à Berkeley une passerelle fut établie entre le réseau ARPANET, qui reliait l'ensemble des établissements en contrat avec le Département de la Défense, et Usenet. En 1985 les messages Usenet adoptèrent le format NNTP qui s'imposa comme le standard de la transmission de *news*, ce qu'il est encore aujourd'hui².

En 1983 la position dominante de AT&T entraîna le découpage des Bell Labs en une série de petites compagnies privées autonomes et le système d'exploitation Unix devint un produit commercial comme un autre. Les universités furent alors confrontées à un problème inattendu : elles durent

1. ME : l'activité académique favorise la création d'écosystèmes.

2. ME : l'émergence de standards ouverts est un terreau favorable au développement économique.

désormais payer leurs licences Unix bien plus cher qu'auparavant¹ pour pouvoir utiliser le système. La réaction des universitaires ne tarda pas et en 1984, Richard Stallman lança le projet GNU avec l'objectif de développer de manière collaborative un système d'exploitation équivalent à Unix qui serait libre, gratuit et modifiable².

Un an plus tard, Stallman créa la Free Software Foundation (FSF) afin de garantir l'ouverture des codes publiés dans le cadre du projet GNU et de les garantir contre le risque d'une appropriation. La FSF énonça les quatre libertés du logiciel libre : liberté d'exécuter le programme pour tous les usages, liberté d'étudier le fonctionnement du programme à travers son code source et de l'adapter, liberté de distribuer des copies, liberté d'améliorer le programme et de le rediffuser accompagné de son code source.

Quelques années auparavant IBM avait sous-traité le système d'exploitation de sa nouvelle gamme de micro-ordinateurs IBM PC à Microsoft. Le grand succès de cette machine généra d'importants revenus pour Microsoft et l'entreprise se lança dans l'écriture d'une couche d'interface graphique destinée à concurrencer le nouveau produit d'Apple, le Macintosh, qui était muni d'une interface graphique fenêtrée avec des icônes et un dispositif de pointage intégré. Les métiers de ces deux entreprises étaient très différents et ces dernières ont déployé des stratégies différentes. Apple, qui amortissait le développement du système d'exploitation de sa machine en intégrant son coût dans le prix de vente de ses micro-ordinateurs, suivait le modèle économique historique de l'informatique en fournissant le système d'exploitation en même temps qu'elle vendait l'ordinateur, donnant ainsi aux clients l'impression que le logiciel système était gratuit. Microsoft, qui ne vendait pas de matériel mais seulement des logiciels qui pouvaient être dupliqués par simple copie, fut alors confronté à un problème inédit. La stratégie de l'éditeur consista à passer des accords avec les constructeurs de matériel afin que son système d'exploitation soit pré-installé sur les machines commercialisées³ soumettant

1. Jusque-là les licences étaient gratuites pour les applications liées à la recherche et très peu coûteuses pour les usages administratifs.

2. ME : la mutualisation de l'effort de développement par effet de levier est une source de croissance pour les entreprises, elle leur permet d'affecter leurs ressources sur leur cœur de métier.

3. ME : la vente liée est controversée, voire interdite, mais elle permet de réaliser d'importants profits.

L'utilisation de ces dernières à l'acceptation d'un « contrat de licence d'utilisateur final » (ou CLUF, en anglais « *end-user license agreement* », EULA). Ce contrat limitait considérablement les droits de l'utilisateur du système en lui imposant une longue série d'interdictions, notamment de revendre sa licence, de réinstaller le logiciel sur une autre machine ou d'en utiliser des copies sur plusieurs machines. Cette pré-installation généralisée des systèmes d'exploitation de Microsoft sur les micro-ordinateurs constitua une barrière à l'entrée efficace qui a longtemps entravé la diffusion de systèmes d'exploitation alternatifs, ce malgré leur disponibilité (DR-DOS, Concurrent CP/M86, MultiUsers Dos, FlexOS 286, etc.).

Malgré la protection apportée par la GPL la communauté GNU peinait à développer le noyau qui lui aurait permis de disposer de son système d'exploitation libre. Pour autant, cela n'entraîna pas une montée en puissance du développement de logiciels libres, et notamment des outils, durant toute la décennie. Les méthodes de développement collectif commencèrent à se formaliser, notamment grâce à l'arrivée d'Internet qui était de plus en plus utilisé pour échanger des données dans le milieu du développement logiciel : on apprenait à travailler à distance, à s'organiser, on développait des outils collaboratifs.

En 1988, Digital Equipment Corporation réunit les grands constructeurs (HP, Bull, IBM, Nixdorf, Siemens, Apollo) autour de l'initiative Open Software Foundation avec l'objectif d'assurer la standardisation d'Unix.¹

Cette volonté des principaux acteurs du monde Unix de travailler ensemble sur un projet de standardisation montre que dans ces grandes entreprises on considérait unanimement une collaboration entre concurrents comme étant favorable aux affaires. Après une longue période pendant laquelle les acteurs tentaient de se distinguer les uns des autres avec des solutions volontairement différenciées, la standardisation est apparue comme un atout.²

IBM décida alors de développer AIX, sa version d'Unix, et HP la sienne, HP-UX. Au même moment l'IEEE spécifiait le standard d'API Posix pour tenter de rendre les différentes variantes d'Unix plus interopérables. On

1. ME : premier modèle à fondation, rassemblant des concurrents autour d'un projet commun.

2. ME : la standardisation permet de favoriser l'émergence d'écosystèmes propices aux affaires.

voit apparaître ici une séparation entre des couches basses (le système d'exploitation avec ses services de base) pour lesquelles les concurrents n'hésitaient pas à collaborer en mutualisant leurs efforts, et les couches supérieures (ergonomie, richesse fonctionnelle, etc.) qui constituaient le lieu de la concurrence commerciale. Les principes énoncés par la FSF et appliqués par le projet GNU donnèrent lieu en 1989 à la création de la General Public Licence ou GNU GPL qui formalisa les droits et devoirs des créateurs et des utilisateurs du logiciel en défendant l'idée que l'utilisateur doit pouvoir contrôler les programmes et non le contraire. Ce recours à la dimension juridique pour encadrer l'utilisation du logiciel à travers une licence fut une étape déterminante pour le logiciel libre.

Ainsi la décennie 1980 a vu émerger des solutions alternatives à Unix : Microsoft Windows et l'IBM PC, le Macintosh d'Apple et les prémices du logiciel libre (outils de développement GNU, etc.). Le modèle de vente liée (CLUF/EULA) et de vente d'ensembles représentait alors une part importante du marché. Le logiciel libre en était à ses débuts, la priorité des développeurs étant le développement de logiciels d'infrastructure et d'outils (réseau, *middleware*, bases de données, serveurs). Les solutions propriétaires furent progressivement remplacées par des logiciels libres dans un mouvement partant des infrastructures et orienté vers les couches applicatives.

3. Années 1990, naissance de GNU/Linux : du logiciel libre à l'Open Source Initiative (OSI)

L'un des évènements marquants du début de la décennie 1990 fut la naissance du World Wide Web, dont Tim Berners Lee, chercheur au CERN décida de libérer les logiciels :

D'abord, il n'est pas évident que je serais devenu un homme très riche si j'avais fait breveter une chose ou deux, parce qu'il y a plein de petits projets hypertexte qui n'auraient pas décollés, dont une bonne partie à cause de l'existence d'un contrôle central. Si j'avais été la personne centrale auprès de laquelle vous auriez dû enregistrer chacun de vos clics, payer pour chacun de vos clics, ou enregistrer chacune de vos pages web alors j'aurais eu un *business model*. Mais

aucun business n'aurait existé. Il n'y aurait pas eu de Web. À la place il y aurait eu plein de gens qui auraient créé plein de Webs incompatibles, ils auraient tenté de contourner mes brevets, donc ils auraient fait des systèmes qui n'auraient pas fonctionné de la même façon¹.

En février 1992, Linus Torvalds diffusa son noyau Linux sous licence GPL. Les développeurs affluèrent pour contribuer et dans le courant de l'année les premières distributions firent leur apparition. C'est à ce moment que furent créées les premières entreprises privées dont le modèle économique était basé sur le logiciel libre. La société Cygnus Solutions présentée comme pionnière en la matière proposa des services de support des outils de développement GNU. Michael Tiemann, fondateur de la société, a fait le pari qu'un marché existait désormais en s'inspirant du GNU Manifesto dans lequel Stallman suggérait que des activités de consulting et de services autour du logiciel libre pouvaient tout à fait permettre de gagner sa vie. Stallman partait du principe que le logiciel à code ouvert engendre un marché de services d'une qualité bien supérieure pour les clients que l'offre monopolistique associée aux logiciels propriétaires. Après s'être convaincu qu'il était possible de proposer des prestations avec une efficacité de deux à quatre fois supérieure à une prise en charge interne, et cela pour un coût de deux à quatre fois inférieur, Tiemann lista quelques questions clés permettant d'évaluer les chances de succès d'un modèle :

- estimer la valeur qu'il est possible de produire ;
- vérifier que le modèle passe à l'échelle ;
- préciser le niveau de problème que l'on est en mesure de traiter ;
- estimer le rythme d'innovation que le projet peut soutenir².

Au cours de la décennie 1990, les distributions GNU/Linux continuèrent à apparaître : la Debian en 1993, puis la Red Hat en 1994³. Alors que la diffusion du logiciel libre se limitait jusque-là à des distributions de

1. Tim BERNERS-LEE, *The writing code, Stata Center Opening Symposium*, 2007, URL : <http://www.thewritingcode.com/pdfs/TimBLTranscript.pdf>. ME : un choix doit être fait entre velléités de contrôle et le potentiel de diffusion d'un projet d'*open source*.

2. Michael TIEMANN, *History of the OSI*, 2012, URL : <http://www.opensource.org/history>.

3. ME : édition de distributions de logiciels à valeur ajoutée (installateur, intégration, cohérence et compatibilité des versions, assurance juridique, etc.).

GNU/Linux intégrant des utilitaires système, des logiciels applicatifs apparurent : un projet de moteur de base de données libre MySQL, un SGBD présenté par ses concepteurs comme « simple, rapide et fiable » et le projet Apache HTTPD, un serveur HTTP modulaire supportant des sites web complexes. Contrairement à ses concurrents (Iplanet de Netscape et IIS de Microsoft) Apache était téléchargeable avec son code source. Il proposait au départ les mêmes fonctions que ses concurrents mais l'affluence des contributions de la communauté lui a rapidement apporté une richesse fonctionnelle et une fiabilité sans équivalent. Les entreprises et les FAI ayant bien compris l'intérêt à la fois technique et économique que représentait l'intégration de ce logiciel au sein des solutions Web qu'ils proposaient, Apache s'est imposé comme la brique standard pour la fourniture de services Web. L'omniprésence d'Apache a drainé de nombreux utilisateurs vers GNU/Linux et a contribué à faire reconnaître la qualité des logiciels applicatifs issus du mouvement du logiciel libre.

En 1995, la massification d'Internet engendrant une forte demande de connexion, l'on assista à l'explosion du nombre de fournisseurs d'accès. Les entreprises commencèrent à prendre la mesure de l'importance de cette infrastructure nouvelle et de l'opportunité qu'elle représentait : il s'agissait d'un nouveau média de promotion et de vente pour leurs services et leurs produits. Elles construisirent leurs services en ligne sur la base du serveur Apache qui a largement contribué au décollage à la fois Web et du logiciel libre et accompagné la massification d'Internet. De ce fait il fut l'un des principaux vecteurs de l'e-commerce.

Début 1997, Eric Raymond écrivit son fameux texte *La cathédrale et le bazar* dans lequel il proposait un modèle explicatif de l'efficacité du développement du logiciel libre et énonçait les raisons pour lesquelles il surpassait le modèle du logiciel propriétaire ¹.

Les arguments de Raymond ont été régulièrement repris par les uns et les autres, parfois pour prendre des décisions majeures, comme Netscape qui, pour des raisons stratégiques, ouvrit le code source de son navigateur, donnant lieu à la création de la Mozilla Foundation dont le navigateur Internet, Firefox, est aujourd'hui le seul véritable concurrent d'Internet

1. Eric S. RAYMOND, *The Cathedral and the Bazaar*, Sebastopol, CA : O'Reilly, 2001, URL : <http://www.catb.org/esr/writings/homesteading/>.

Explorer¹. Le texte de Raymond, qui venait compléter l'approche philosophique de Stallman en présentant une perspective plus pragmatique, proposait un argumentaire indiquant aux entreprises ce qu'elles pouvaient gagner à intégrer le logiciel libre dans leur stratégie. La segmentation et la spécialisation des licences (Apache, MIT, FreeBSD, etc.) ouvrirent de nouvelles possibilités aux entreprises. Durant cette évolution la licence commença à jouer un rôle d'articulation entre le logiciel et son potentiel économique et/ou de diffusion. Les années 1990 marquèrent aussi l'émergence de la communauté des développeurs en tant que composante essentielle d'un projet de logiciel libre et comme un facteur clé de son succès. Ross Gardler, responsable de la communauté des développeurs Apache², utilise l'expression « *community over code* » pour illustrer la primauté de la qualité de la communauté sur celle du code.

Comme le mouvement du logiciel libre prenait de l'ampleur, les libristes qui le portaient cherchèrent à en faire une activité qui puisse leur permettre d'en vivre pour s'y consacrer à plein temps. Les modèles qu'ils trouvèrent pour cela les amenèrent invariablement vers le marché des entreprises privées. Comme le mot *free* de *free software* paraissait inquiéter les investisseurs et les clients, un groupe d'acteurs du mouvement constitué de Michael Tiemann de Cygnus Solutions, Bruce Perrens de Debian, et Linus Torvalds décidèrent, avec l'aide d'Eric Raymond et de quelques autres, d'utiliser désormais la formule *open source software* ou logiciel à code source ouvert. Ce terme fut diffusé sur le Web en 1997 et l'année suivante, l'Open Source Initiative (OSI) fut créée avec pour mission de sensibiliser le public et le secteur privé de l'importance et de l'intérêt du logiciel non propriétaire.

À ce titre, la première distribution GNU/Linux française, Mandrake (qui deviendra plus tard Mandriva), fut créée en 1998. Le modèle économique adopté par Mandrakesoft, sa société éditrice, était celui de la distribution à valeur ajoutée : les souscripteurs recevaient un CD qui contenait plus de fonctions, par exemple une compilation de drivers beaucoup plus large que la version téléchargeable gratuitement, ainsi qu'un accès privilégié au support.

1. ME : céder du contrôle vs gagner de l'ampleur.

2. ROSS GARDLER, « Community over code », dans : *Conference fOSSa*, Grenoble, 2010.

En 1999, le succès d'Apache amena la communauté à créer une fondation, financée par des dons et la vente de produits dérivés, dont l'objectif était d'apporter un cadre légal aux projets qu'elle hébergeait. Uniquement basée sur le volontariat, elle défendait, en tant que personne morale, les contributeurs et les utilisateurs contre d'éventuelles poursuites liées à des questions de propriété intellectuelle. De plus, le noyau historique de la communauté s'employait à éviter les dérapages. « Même très impliqué, un développeur ne peut pas faire n'importe quoi, car tous les votes sont publics » expliquait Sylvain Wallez à ce propos, « il est donc très difficile d'abuser du système pour imposer une modification à des fins purement commerciales. »¹

À la fin de la décennie 1990, l'éditeur norvégien eZ Systems AS proposa eZ Publish, le premier système de gestion de contenu (CMS ou *content management system*). Il s'agissait d'un serveur permettant de publier facilement des contenus, ces derniers étant stockés dans une base de données. La conception des pages s'apparentait à une configuration et ne nécessitait aucune compétence particulière en HTML. L'originalité du logiciel résidait dans l'association d'un système d'exploitation GNU/Linux, d'un serveur Web Apache, d'une base de données MySQL et d'un service de création de contenu basé sur le langage de script PHP. Cette architecture désormais appelée « LAMP » s'est ensuite imposée comme un standard, non seulement pour les CMS, mais aussi dans d'autres domaines : plateformes collaboratives, e-commerce, etc. eZ Systems AS proposait son logiciel en deux versions, une sous licence GPL et une propriétaire avec un abonnement. Dans le premier cas, le support était obtenu auprès de la communauté et dans le second il était fourni par la société.²

Un exemple illustratif de la professionnalisation de l'*open source* est celui de la société Digium créée en 1999 pour profiter de l'arrivée de la voix sur IP (VoIP) en proposant des alternatives à très faible coût (l'objectif affiché était de proposer des solutions 80% moins chères que les prix habituellement pratiqués) sur le marché particulièrement fermé de l'autocommutateur téléphonique (PABX). La stratégie adoptée pour cela par

1. Cité par Frédéric BORDAGE, « Open source : la naissance d'un écosystème », dans : *01Net* (2005), URL : <http://pro.01net.com/editorial/270447/open-source-la-naissance-dun-ecosysteme/>.

2. ME : modèle de la double licence.

Digium fut d'initier un projet d'autocommutateur bénéficiant de la banalisation des ordinateurs personnels, des distributions GNU/Linux - alors en tête des systèmes d'exploitation de serveurs - et du modèle de développement *open source*. La société s'est aussi engagée dans le développement et la production d'interfaces de téléphonie permettant le raccordement des autocommutateurs à base de PC sur les réseaux de téléphonie existants (RTC, RNIS alias ISDN).

Digium sponsorise et anime la communauté des développeurs et des utilisateurs du système de téléphonie *open source* Asterisk. Son modèle se base sur une offre de solution avec laquelle elle se positionne à la fois sur le volet logiciel (PABX logiciel, drivers, etc. publiés sous licence GPL), sur le volet matériel (gamme de cartes et de boîtiers d'interfaçage analogique et numérique, postes téléphoniques, solutions complètes, etc.) et sur le volet service (intégration, configuration, support, *consulting*, etc.). Asterisk est désormais la référence dans le domaine des PABX *open source* et il couvre une gamme complète d'équipements et une large cible d'utilisateurs. Sa communauté est la plus importante et la plus active dans le domaine de la VoIP (elle compte plus de 8 000 développeurs actifs, les forums Asterisk couvrent 35 000 sujets traités et comptent plus de 120 000 *posts*, et les solutions sont déployées dans 170 pays). Souvent mise à l'honneur par des revues, des salons et des groupements professionnels, Digium a réalisé un parcours exemplaire : une entreprise lance un projet de logiciel libre qui apporte une réponse à un problème largement partagé (un PABX générique robuste sans risque d'appropriation grâce à une licence GPL) et, grâce à une gouvernance éclairée et à des choix pertinents, elle fait d'Asterisk un logiciel de référence¹. Cela a permis l'émergence d'un écosystème de plus en plus riche dans lequel l'entreprise trouve des sources de revenus en vendant du service, du matériel et des solutions complètes. Les responsables de Digium avaient une vision claire du modèle « infrastructure / applications » et de l'intérêt qu'il était possible de tirer de la publication d'un code en GPL pour initier une communauté très impliquée autour d'un code non appropriable. Aujourd'hui l'entreprise pratique son cœur de métier sur les marchés ouverts par l'écosystème qui s'étend de plus en plus largement. Ainsi on assistait en 2012 à l'arrivée dans l'écosystème d'un acteur proposant une solution *openhards*

1. ME : création réussie d'un écosystème riche et varié.

ware (XiVO IPBX) dédiée aux télécommunications et capable de traiter plus de mille communications simultanées.

Comme nous venons de le voir les années 1990 amenèrent leur lot d'innovation avec le serveur web Apache qui a contribué à donner une image positive au logiciel libre et permis à de nombreux FAI de disposer à moindre coût d'une offre de services autour du Web. Internet est devenu un réseau de communication de masse et l'e-commerce a fait son apparition. La nouvelle terminologie *open source* a donné au mouvement une crédibilité auprès des acteurs industriels.

4. Première décennie 2000, professionnalisation de l'*open source* et structuration d'un nouveau secteur économique

Le début des années 2000 fut un moment clé pour le logiciel libre. L'on assista à une forte professionnalisation du mouvement et l'on vit le logiciel libre qui, après avoir investi le domaine des infrastructures (par ex. GNU/Linux, Sendmail, Apache), se diriger vers les couches applicatives en investissant successivement les bases de données (par ex. PostgreSQL, MySQL), le *middleware* (par ex. Jboss), les langages de développement web de services riches (par ex. PHP) et les couches applicatives (par ex. Ez Systems, Drupal, etc.).

Un cas emblématique de cette tendance est la lame de fond de l'architecture LAMP qui, partant d'une combinaison de quatre logiciels très différents, mais présents dans toutes les distributions, devint la plate-forme standard pour le développement d'applications web ¹.

Cette époque marqua également l'arrivée sur le marché d'entreprises d'un type nouveau : les SS2L ou sociétés de service en logiciel libre. Généralement ces entreprises n'éditent pas de logiciels libres mais elles contribuent activement à des projets, font des développements complémentaires et de l'intégration et vendent des services à des clients utilisateurs (les différents rapports de la Linux Foundation indiquent que le développement des mises à jour du noyau est assuré par une majorité croissante d'entreprises). La société Linagora est un exemple de SS2L qui s'est

1. ME : vente de services autour d'une base standardisée.

lancée en 2000 avec des conventions de formation pour l'armée de l'air et un contrat de développement d'un logiciel avec la mairie de Besançon.

En 2000, MySQL AB troqua la licence LGPL utilisée dans sa base de code pour la licence GPL. Ce changement eut un fort impact tant au niveau communautaire que financier. Le choix de la licence GPL a pour effet de débrider la collaboration et de contrôler la concurrence (les concurrents qui téléchargent un logiciel sous licence GPL ne peuvent l'améliorer et en tirer des profits sans restituer ces améliorations à la communauté). La même année vit la création de la société Jboss, un *middleware*, Java, j2EE qui devint extrêmement populaire chez les développeurs d'applications web.

2001 fut l'année de naissance du CMS Drupal développé par des étudiants en fin d'études qui décidèrent de se lancer sur un projet de gestion de contenu web, simple, modulaire et accessible à des non informaticiens. Au bout d'un an, le projet Drupal fut publié sur la toile où il se diffusa rapidement. La société Acqua fut créée pour vendre des services autour de Drupal alors proposé en deux versions, la première étant disponible pour la communauté (gratuite, code source ouvert) et la seconde payante, mais proposant les dernières fonctionnalités. L'originalité du mécanisme de licence réside dans le fait que la version payante est transférée à la communauté dix-huit mois après sa publication¹.

À cette époque *Forbes magazine* annonçait près de douze millions d'utilisateurs pour GNU/Linux qui devint le deuxième système d'exploitation sur le marché des serveurs avec 27% du marché derrière Windows (41%). Apache possédait pour sa part 51% du marché des serveurs web.

En 2002, près de 60% des développeurs déclaraient vouloir réaliser des applications pour GNU/Linux dans l'année suivante, et le taux de pénétration de ce système d'exploitation dans les entreprises japonaises bondissait de 35 à 65% en un an. C'est à ce moment que fut créée la fondation ObjectWeb. Son objectif était de proposer des composants *open source* dans les couches intergiciels. Cette même année vit la naissance de l'association Adullact dont la mission était d'identifier et d'exprimer

1. ME : la licence différée est un modèle répandu. Elle présente cependant l'inconvénient, en restreignant la diffusion des versions récentes, de proposer paradoxalement aux utilisateurs payants un logiciel moins testé donc de moindre qualité que ne l'est la version ouverte.

les besoins des collectivités territoriales, administrations, centres hospitaliers, etc. « afin de mutualiser et maintenir un patrimoine commun de logiciels libres utiles aux missions de services publics » (selon le site web d'Adullact)¹. Cette approche du logiciel libre à l'initiative d'une fédération utilisateurs/clients était nouvelle et présentait un double intérêt : elle permettait de générer d'importantes économies par mutualisation – on y reconnaît l'effet de levier propre à l'*open source* – et constituait en même temps, de manière cumulative, un patrimoine disponible pour toute la communauté d'utilisateurs².

Hewlett Packard disposant d'un gros vivier de partenaires vit également dans le logiciel libre un gisement de revenus potentiels à exploiter. La firme construisit une offre de services de support autour de logiciels *open source* en proposant des solutions clefs en main comprenant des serveurs web, des bases de données et des outils *middleware* tels que Jboss (qui débuta en vendant la documentation de ses logiciels), cela sur des machines serveurs de la marque exploitées sous GNU/Linux. Des partenariats furent établis avec des sous-traitants pour proposer du support de niveau 1 et 2, le support de niveau 3 étant généralement confié à l'éditeur du logiciel supporté.

4.1 Eclipse et son écosystème

On assista de plus en plus souvent au début des années 2000 à des stratégies de constitution d'écosystèmes. Alors que certaines structures comme la fondation Eclipse, orientées vers le développement, se focalisaient sur les processus et sur les *workflows* associés, d'autres comme Apache étaient plutôt tournées vers la création et le développement de leur communauté. Lorsqu'elles atteignaient une masse critique, les infrastructures issues de ces projets constituaient des environnements propices à l'émergence d'une économie de services : installation, formation, support, développements spécifiques, vente et maintenance de matériel, etc.

1. ME : effet de levier par mutualisation de ressources, plus le problème est partagé entre un nombre élevé d'utilisateurs, plus cet effet est important.

2. François ÉLIE, « Mobile, Cloud and the Digital Native Generation », dans : *Conferrence fOSSa*, INSA Lyon, 2011. ME : contrôle du projet par les utilisateurs pour une meilleure prise en compte des besoins.

L'ensemble constitué par le logiciel libre et le foisonnement de services proposés autour formaient un écosystème¹.

En Amérique du Nord au début de l'année 2004, plus d'un million de développeurs travaillaient sur des projets de logiciels libres, 67% des entreprises utilisaient des logiciels libres et MySQL progressait de 30% soit cinq fois plus rapidement que Microsoft SQL Server. Le concept d'écosystème, qui dans ce contexte prend tout son sens, jouait désormais un rôle déterminant à travers l'établissement et la consolidation de standards, la création d'un marché ouvert et concurrentiel, créant ainsi une stabilité très prisée par les acteurs industriels qui étaient incités à y participer en tant qu'utilisateurs et parfois également comme contributeurs. Les participants qui contribuaient à l'écosystème bénéficiaient d'un accès privilégié.

La pérennité de ces écosystèmes a néanmoins un prix : celui d'une gouvernance adéquate pour gérer l'architecture de collaboration commune et maintenir l'adéquation de l'infrastructure, clé de l'écosystème, avec les attentes des acteurs. La communauté existante doit rester très impliquée et gagner de nouveaux membres pour augmenter l'effet de levier et faire évoluer l'infrastructure en termes de qualité et de fonctionnalités. Cette exigence d'évolution est souvent la raison qui justifie l'ouverture de code. Ce fut le cas de Java, ouvert par Sun Microsystems, de la fondation Eclipse, créée par IBM pour disposer d'une plate-forme de développement qui manquait dans l'environnement GNU/Linux, ou encore de Netscape pour garantir la stabilité des standards ouverts du Web qu'une situation hégémonique de Microsoft avec Internet Explorer aurait pu remettre en question.

Publier un logiciel sous licence libre augmente sa diffusion, son utilisation et donc la taille du marché potentiel de services, d'équipements et de conseil dans l'écosystème qui s'est créé autour de ce logiciel.

Les acteurs jouant un rôle prédominant dans un écosystème augmentent leur contrôle sur la concurrence. Ainsi la création de la fondation Eclipse par IBM avait également pour objectif, à travers la mise au point d'un environnement de développement à la fois performant et accessible, de favoriser l'utilisation d'équipements informatiques basés sur les architectures x86 en faisant croître la base de machines exploitées sous GNU/Linux et

1. ME : Modèle à fondation tourné vers la création d'un écosystème.

le parc d'applications, et d'entraver comme cela ses principaux concurrents (Microsoft, Sun Solaris).

Cet écosystème a permis à IBM de créer une forte dynamique basée sur une plate-forme valorisée à plus de 40 millions de dollars et sur un large réseau d'entreprises. L'environnement Eclipse permet aux éditeurs de logiciel quelle que soit leur taille – mais aussi aux étudiants – de disposer d'un environnement au dernier état de l'art pour développer des applications pour GNU/Linux sous langage Java. IBM a gagné son pari puisque, avec plus de 10 000 téléchargements par jour, Eclipse s'est désormais imposé comme ce que les Anglais désignent sous le terme de *commodity*. Ce mot sans équivalent simple en français représente une matière première, un produit, une ressource générique dont le choix s'impose naturellement.

Les écosystèmes peuvent s'apparenter à des marchés, permettant de dégager beaucoup de valeur lorsqu'ils sont ouverts mais souffrant des tentatives de contrôle, jusqu'à pouvoir disparaître. À un certain stade de son développement Eclipse a connu une crise : la volonté d'IBM de contrôler l'écosystème de manière directive a provoqué le départ de contributeurs qui avaient l'impression de ne plus être au service d'un projet d'infrastructure de développement ouverte mais de travailler pour servir les intérêts d'IBM. La stagnation de la communauté a poussé la fondation Eclipse à couper les ponts avec son sponsor historique et retrouver une certaine neutralité aux yeux des membres de la communauté qui, aussitôt le divorce consommé, a recommencé à prospérer¹. Une situation similaire, et qui s'est mal terminée, est celle de Nokia qui, à force de trop vouloir le contrôler, a fini par tuer l'écosystème Symbian, et finalement le système d'exploitation Symbian lui-même. L'écosystème est un terrain qui offre de nombreuses opportunités, par exemple la société O'Reilly, créée en 1984, initialement tournée vers Unix, est aujourd'hui l'éditeur de référence de la littérature technique sur le logiciel libre.

4.2 Les solutions d'entreprises

Les entreprises sont utilisatrices d'une grande variété de logiciels qui vont du *desktop* jusqu'à des applications industrielles et commerciales :

1. ME : autre illustration du compromis contrôle vs ampleur.

ERP et CRM (*Entreprise Ressource Planning* et *Customer Relationship Management*) mais aussi *supply chain*, *KM knowledge management systems*, *GroupWare*, *e-learning*, etc. Les principaux acteurs de ce secteur sont IBM, BAAN, SAP et Oracle.

La hauteur des enjeux, la nécessité d'adapter les solutions et les processus à l'organisation et aux spécificités des entreprises, l'exigence d'un support sans faille et les questions de responsabilité ont fait que les solutions propriétaires ont longtemps régné en maîtres dans ce domaine. Les contraintes strictes exigées par les industriels, notamment la question de l'identification des responsabilités, et le caractère stratégique de ces applications ont probablement retardé l'arrivée du logiciel libre sur ces marchés. Est-ce dû à une évolution des mentalités ou à une recherche d'économie, toujours est-il qu'au milieu de la décennie les solutions d'entreprises basées sur des logiciels libres ont commencé à faire leur apparition. Pour citer deux exemples populaires, Open ERP est un logiciel intégré d'entreprise et SugarCRM un logiciel de gestion de relation clients qui suit le modèle de double licence comme MySQL à ceci près que la version commerciale supporte des fonctions supplémentaires. Le logiciel libre, qui inquiétait de moins en moins les décideurs et qui présentait un avantage financier indéniable, a finalement eu raison de leurs réticences. Une étude de McKinsey a montré que la structure de coût d'installation d'une solution ERP ou CRM propriétaire est composée d'environ 30% pour la licence et de 70% pour le développement des adaptations.

Au fil des ans les solutions d'entreprises basées sur des logiciels libres ont bénéficié de gros enrichissements fonctionnels et leur ergonomie a évolué significativement. SugarCRM et Open ERP ont chamboulé un secteur partagé entre les grands éditeurs informatiques. Cette adoption du logiciel libre par les entreprises a touché tous les composants du système d'information, ainsi une étude menée en février 2005 par Evans Data auprès de développeurs et d'administrateurs de bases de données montrait que 64% des SGBD installés en entreprise étaient *open source*. Fin 2005, une étude d'Optaros montrait que 87% des entreprises utilisaient des logiciels libres et BusinessWeek annonçait que le logiciel libre avait « franchi le seuil critique ». Ce phénomène est parfois évoqué comme la « rupture des digues » qui avaient un temps isolé les entreprises du mouvement du logiciel libre.

Les *success-stories* que nous évoquons ici ne doivent pas faire oublier que tous les projets de logiciel libre ne sont pas des succès et d'autre part le fait que bien souvent la visibilité des projets intervient longtemps après leur lancement. On peut estimer cette durée de gestation à 7 ou 10 ans en moyenne. Bien souvent les initiateurs de ces projets les ont démarrés dans une chambre d'étudiant ou menés de front avec une autre activité professionnelle, ce parfois durant plusieurs années. On parle parfois de projets « lancés dans un garage ». Des projets comme MySQL, Nagios et même GNU/Make ou GNU/Emacs, ont suivi cette longue gestation. Il existe cependant des exceptions, comme le système de gestion électronique de documents Alfresco qui, lancé en 2005, a connu un succès quasi immédiat. Cette réussite très rapide peut s'expliquer de plusieurs façons. Le logiciel apportait une réponse à un besoin bien identifié car parfaitement connu du fondateur John Powell (cofondateur de Documentum), et permettait aux entreprises de réaliser d'importantes économies. Cependant un élément probablement déterminant fut de bénéficier d'un lancement de type *startup*. Cet exemple illustre l'importante mutation du logiciel libre qui dans la première décennie des années 2000 commence à brûler l'étape du garage pour passer directement par la banque.

4.3 Quelques remarques sur le bazar

Le début des années 2000 a marqué le véritable démarrage du logiciel libre. Les créations d'entreprises se sont succédé et de nombreux projets de logiciels libres lancés quelques années auparavant sont parvenus à maturité et leur diffusion fut suffisamment large pour qu'apparaissent des écosystèmes de services, de conseil et de support dans toutes sortes de domaines. Ce mouvement s'accompagna d'une spécialisation (Apache pour l'infrastructure, OW2 pour les intergiciels, Eclipse pour le développement de code, Funambol pour le développement et déploiement d'application sur terminaux mobiles, etc.). Avec la montée en taille des projets, de plus en plus d'acteurs du logiciel libre recherchèrent un moyen de vivre de cette activité. On vit alors émerger de multiples modèles économiques basés sur (ou combinant) :

- une offre commerciale de services (*consulting*, support, formation, etc.) ;

- des logiciels à double licence (une version libre et une version commerciale) ;
- la vente de distributions à valeur ajoutée (distributions GNU/Linux packagées, testées, qualifiées) ;
- des logiciels à licence différée (la version courante est vendue, les versions antérieures sont libres et gratuites) ;
- une mutualisation (création d'un effet de levier afin de partager les coûts de développement d'un logiciel).

Les entreprises qui se spécialisèrent dans la prestation de service autour des logiciels libres sans être nécessairement les auteurs du code (mais y contribuant souvent) furent baptisées sociétés de service en logiciels libres ou SS2L.

La seconde moitié de la décennie 2000 vit l'émergence de projets de logiciels libres basés sur un *business plan*, à la façon de MySQL en son temps, mais en disposant dès le départ, avant même qu'une ligne de code ait été écrite, d'un budget d'investissement et de fonctionnement. En somme ce sont des *startup* de l'*open source* dont l'objectif est de lancer très rapidement le logiciel libre afin d'occuper le créneau et de créer un écosystème dont l'entreprise qui l'a lancé puisse tirer profit.

Après avoir conquis les couches basses de l'informatique, le logiciel libre s'orienta vers les couches applicatives et vers les solutions d'entreprises (ERP, CRM, GED) ce mouvement s'accompagnant de la création de sociétés de service fournissant des prestations autour de logiciels libres.

La multiplication des projets de logiciel libre fait que les métiers dans lesquels une solution de logiciel libre n'existe pas se font rares.

4.4 Dans les nuages

On vit réapparaître des offres de services basées sur la technologie client-serveur en vogue durant les années 1960 à 1990 : les SaaS (*Service as a Software*), les IaaS (*Infrastructure as a Software*), les PaaS (*Platform as a Service*) et les DaaS (*Desktop as a Service*). La plupart de ces services étaient basés sur des briques *open source* et organisés sur le modèle des services en ligne. Si la solution semble claire pour le client qui paie pour obtenir un service, elle l'est moins pour l'éditeur du service dont

les revenus sont générés en partie grâce à l'exploitation de logiciels libres développés par d'autres.

L'extension du logiciel libre l'a amené à investir de nouveaux terrains de jeux, notamment les smartphones et les tablettes qui ont la particularité de constituer une plate-forme destinée à recevoir des logiciels propriétaires (*apps*). Ces appareils, livrés avec un système d'exploitation pré-installé, ont ré-adopté le modèle de vente d'un ensemble comme il était pratiqué dans les débuts de l'informatique. Apple a montré la voie des applications pour mobiles avec son AppStore, suivi par Google Play, le magasin d'applications de la plate-forme Android. Ce modèle s'apparente à celui du supermarché : le propriétaire de la plate-forme sélectionne les applications, les expose dans ses rayons et prélève une partie du montant des ventes.

Les services en ligne sont également un moyen pour les éditeurs de logiciels libres de générer des revenus à travers des abonnements donnant accès à différents types de prestations (forges, stockage, etc.). Ainsi Ubuntu a intégré la fonction Ubuntu One. Il s'agit d'un logiciel donnant accès à un service de stockage en ligne. Deux gigaoctets sont fournis gratuitement et au-delà de cette capacité le service devient payant. Près de 40% des utilisateurs du système Ubuntu se sont ouvert un compte sur ce service (selon l'étude *Ubuntu user survey* 2012). Par ailleurs la fouille de données et la vente d'informations à destination d'applications marketing sont d'autres moyens, certes plus controversés, de générer des revenus complémentaires.

4.5 Le phénomène de concentration

On assiste depuis quelques années à un mouvement de concentration dans le monde de l'édition de logiciels libres, les petits éditeurs étant absorbés par ceux de taille moyenne, eux-mêmes rachetés par les grands. Une illustration de cette tendance est le rachat de MySQL par Sun Microsystems ensuite racheté par Oracle. Cet exemple appelle plusieurs remarques. Tout d'abord, une opération de ce type a pour objectif essentiel la prise de contrôle d'un logiciel. Ensuite un tel rachat peut être extrêmement intéressant pour un acteur qui voit sa domination menacée par un nouvel entrant, surtout s'il s'agit d'un logiciel libre. Ensuite force est de

constater que le rachat n'est réalisable qu'aux deux conditions suivantes : a) que le logiciel en question soit publié sous une licence de type *copy-left* faible ou *copyright* et b) qu'une entité dispose des droits sur le code par exemple par la signature d'une licence de contributeur (*Contributor License Agreement* ou *CLA*) prévoyant la cession de tous les droits des auteurs vers la structure ou bien par la mise en place de contrats de travail avec les contributeurs. Si cela n'est pas le cas – licence de type *copy-left* fort de type *GPL* – alors l'opération de prise de contrôle est impossible. On peut se demander où s'arrêtera ce mouvement de concentration et s'il a vraiment un sens. L'histoire des modèles économiques du logiciel libre montre que les velléités de fermeture ou de contrôle du code entravent la diffusion, brisent la confiance des contributeurs, affaiblissent l'écosystème et se révèlent finalement contre-productives. En évoquant la scission d'OpenOffice et la création de LibreOffice, Philippe Scoffoni constate qu'il est de plus en plus évident qu'un logiciel libre ne peut pas être porté directement et éternellement par une entreprise, car « un logiciel libre est un bien commun et ne peut appartenir à personne » et que « à ce jour seules les associations et les fondations semblent en mesure de le garantir ». Il ajoute que « l'apport et la contribution des entreprises sont essentiels. »¹

Ces dernières contribuent en effet à l'écosystème en alimentant les marchés qui s'y développent et contribuent parfois activement au développement du code. Un autre type d'acteur est en mesure de renverser cette tendance à la concentration sans toujours en avoir conscience : il s'agit des contributeurs eux-mêmes qui constituent la cheville ouvrière du projet et peuvent refuser de contribuer à n'importe quelles conditions.

4.6 *Openness*

Le concept d'*openness* n'est pas nouveau. Lors de ses recherches sur l'histoire des sciences, l'économiste Paul David l'a utilisé pour définir une caractéristique nouvelle et déterminante de la science à l'époque de la Renaissance. On constate qu'à partir du moment où la science devient ouverte, c'est-à-dire lorsque l'information commence à circuler, que les

1. Philippe SCOFFONI, *LibreOffice, OpenOffice un fork vu de l'intérieur*, 2011, URL : <http://philippe.scoffoni.net/libreoffice-openoffice-fork-vu-de-linterieur>.

scientifiques interagissent, éditent et diffusent des résultats de leurs recherches, il se met en place une organisation qui vise un accroissement rapide des connaissances¹.

Ce que montre David à travers ses recherches sur l'*open science* est, si ce n'est le lien de causalité, au moins la coïncidence entre l'ouverture des normes et de la technologie, et l'accélération de la production de connaissances, générant de nouvelles normes et de nouvelles avancées technologiques, etc. La trajectoire de ce cercle vertueux déborde largement du milieu industriel vers la société qu'il irrigue de connaissances et de technologies qui se transforment en expérimentations, en inventions, en savoirs-faire et ultimement en une activité économique.

Le modèle de « communauté de scientifiques » proposé par David renvoie aux « communautés de pratiques » qu'Étienne Wenger définit comme des groupes de personnes en interaction, rassemblées autour d'un même domaine et partageant des pratiques pour résoudre un problème partagé. Wenger a montré que l'une des spécificités de cette structure était de régler les problèmes en produisant des connaissances nouvelles et de les transférer dans la pratique, ce qui fait de la communauté un acteur essentiel de l'innovation.²

Reprenant le modèle de David, les chercheurs Frédéric Foray et Liliane Hilaire-Perez ont mené des investigations historiques et montré comment au XVIII^e siècle la ville de Lyon était parvenue, grâce à la mise en œuvre de ce que les auteurs qualifient d'*Open technology*, à remporter le bras de fer économique qui opposait Lyon à Londres dans la bataille de la soie. Cette confrontation économique est intervenue peu après l'adoption du système de brevet par l'Angleterre alors qu'il n'avait pas encore été adopté en France. Pour piloter l'innovation, la ville de Lyon avait mis en place un dispositif consistant à organiser des réunions lors desquelles les inventeurs présentaient publiquement les améliorations pour les métiers à tisser³. Les industriels présents à ces séances étaient invités, s'ils adoptaient l'une des nouveautés sur leurs métiers à tisser, à indiquer la

1. Paul A. DAVID, *An Essay on Patronage, Reputation and Common Agency Contracting in The Scientific Revolution*, Stanford : Stanford University, 2007, p. 15.

2. Étienne WENGER, *Communities of Practice : Learning, Meaning, and Identity*, 1^{re} éd., Cambridge, Mass. : Cambridge University Press, 1999.

3. ME : modèle de l'*open science* visant à faire circuler l'information le plus librement possible.

nature de la modification effectuée et le nombre de machines modifiées¹. La ville tenait une comptabilité de l'adoption des innovations des différents inventeurs qui étaient récompensés financièrement par la ville². Le fait que les inventeurs soient rémunérés par la collectivité et non par les industriels eux-mêmes montre que l'on avait parfaitement conscience à l'époque de ce que la boucle de retour sur investissement dépassait les seules communautés des inventeurs et des industriels pour concerner la société lyonnaise dans son ensemble³. Celle-ci était très impliquée dans les industries de la soie qui, à la fin du XVIII^e siècle, employaient un quart des 143 000 habitants de la ville.

À Londres l'individualisme, le secret et les brevets mirent un terme à la diffusion de l'innovation alors qu'à Lyon « les règlements locaux transfèrent directement les innovations majeures dans le domaine public de la manufacture ».⁴

Pour évaluer les récompenses accordées aux innovateurs, les autorités locales utilisaient des méthodes assez élaborées. Ainsi, par exemple, un certain Michel Berthet se vit récompensé de 1 000 livres françaises de l'époque. 600 lui furent versées pour la révélation de son invention et les 400 complémentaires après qu'il eut expliqué aux industriels comment la mettre en œuvre « et qu'elle ait été installée dans quatre autres fabriques que la sienne ». Les rémunérations ne récompensaient donc pas seulement l'utilité économique potentielle de l'invention mais elles étaient aussi « indexées sur les efforts de l'inventeur pour partager ses connaissances avec l'ensemble de la communauté ». Les résultats de cette politique de soutien à l'innovation furent remarquables : en 1814, Lyon et Londres disposaient d'outils de production de tailles comparables avec respectivement 14 500

1. ME : mesure de l'impact par un comptage (équivalent du nombre de téléchargements ?).

2. ME : le développement de l'écosystème est une priorité, il est considéré comme la finalité de l'openness.

3. ME : *openness* généralisée → émergence de gros écosystèmes → large impact sociétal → amorçage d'un cercle vertueux de prospérité (production d'une richesse partagée par la grande majorité de la population, etc.)

4. Cité par Dominique FORAY et Liliane Hilaire PEREZ, « The economics of open technology : collective organization and individual claims in the "fabrique lyonnaise" during the old regime », dans : *Frontiers in the economics of innovation and new technology. Essays in honor of Paul A. David*, sous la dir. de Cristiano ANTONELLI et al., Cheltenham : Edward Elgar Pub., 2005, p. 239–254.

et 12 000 métiers à tisser. En 1853, Londres n'en comptait plus que 5 000 alors que la ville de Lyon en exploitait 30 000 (60 000 en comptant les métiers de la banlieue). Ce succès fut durable puisqu'au début des années 1900 la production française avait beaucoup augmenté, la majeure partie étant exportée, alors que l'Angleterre était devenue un importateur net.

Comment expliquer le succès lyonnais ? Foray et Perez considèrent que ce qui a servi Lyon et manqué à Londres fut une flexibilité de la production à destination de l'international, associée à une grande « mobilité organisationnelle » qui produisait une « économie de la variété » basée sur une mobilisation de ressources humaines et techniques, de compétences, de réputation et d'estime de soi¹. Le modèle lyonnais était celui d'une production à la demande qui fournissait des échantillons, en attente des commandes des clients et capable de mobiliser en quelques jours cinq mille métiers à tisser pour livrer rapidement de grosses commandes imprévues². Au contraire, l'industrie londonienne était devenue peu réactive et son modèle avait évolué vers la spéculation, l'anticipation et le stockage³. En outre, là où les acteurs lyonnais étaient en mesure de travailler avec plusieurs partenaires et d'en changer du jour au lendemain, les producteurs londoniens étaient généralement sous la domination des marchands qui leur faisaient des avances de trésorerie puis leur imposaient des contrats d'exclusivité⁴, les plaçant dans une situation d'endettement et de dépendance qui les dissuadaient d'envisager une modernisation de leur outil de production.

Un autre point à souligner fut le rôle important que joua le tribunal des prud'hommes en imposant à tous les fabricants de métiers à tisser des standards sur la forme et les dimensions des composants, permettant l'interopérabilité des pièces détachées, l'augmentation de la polyvalence des mécaniciens et l'optimisation des coûts de maintenance et d'entretien des machines grâce à l'émergence d'un marché de la pièce détachée⁵.

1. ME : la cathédrale.

2. ME : modèle centré sur le client.

3. ME : modèle centré sur le fournisseur.

4. ME : la cathédrale.

5. ME : écosystème de services autour du SAV et de la vente de pièces détachées basé sur la standardisation.

4.7 *Open technology*, logiciel libre et open hardware

Ce voyage dans le temps permet de porter un regard neuf sur le brevet et la propriété intellectuelle mais aussi de découvrir des similitudes parfois frappantes entre le modèle lyonnais de l'industrie de la soie et l'*open source* : les modèles sont similaires de même que les effets produits. On constate également que le modèle de l'*open hardware*, aujourd'hui en fort développement, existait déjà il y a plus d'un siècle. L'*open hardware* consiste à ouvrir les plans, schémas, principes et procédés de fabrication d'objets matériels, en adoptant des principes similaires à ceux du logiciel libre. On constate que l'analogie va loin puisque l'on trouvait déjà rassemblés à l'époque l'ensemble des facteurs clés de succès qui accompagnent la mise en œuvre de la philosophie de l'*openness* : libre circulation de la connaissance, liberté et gratuité d'usage des inventions par les utilisateurs, rémunération indirecte des contributeurs, mise à disposition de l'ensemble de la société d'une infrastructure libre et ouverte. L'*open hardware* concerne par définition des produits rivaux, toutefois ses modèles économiques ne sont pas totalement différents de ceux du logiciel libre puisqu'une partie des revenus provient des services (support, conseil, etc.). L'autre concerne par contre la vente de produits physiques. Dans le domaine de l'électronique par exemple, il s'agit de cartes à microcontrôleurs (par ex. Arduino, Raspberry Pi, etc.) vendues en réalisant une marge sur la fabrication et sur la distribution. Les revenus peuvent aussi provenir de produits dérivés (T-shirts, autocollants, figurines, etc.) à l'instar de ce que font les associations « libristes » qui vendent des goodies à l'occasion des conférences et des salons. La plupart des caractéristiques intéressantes du modèle du logiciel libre restent valides dans le cas de l'*open hardware* : existence d'un écosystème à partir duquel des marchés concurrentiels se développent, système d'entraide très efficace, très faible coût du matériel dont les plans et schémas sont ouverts, écosystèmes logiciels basés sur le libre, etc. Des communautés de l'électronique *open-hardware* très dynamiques se sont constituées autour de projets ou d'entreprises comme Arduino, Linaros, Raspberry pi, Diydrones et d'autres. En 2010, une étude conduite par la société Adafruit Industries a recensé treize entreprises de cette catégorie. Prises dans leur ensemble, elles représentaient un chiffre d'affaires cumulé de cinquante millions de dollars. Robert Viseur, chercheur en gestion de la co-création à l'Univer-

sité de Mons, prévoit que le marché de l'*open hardware* devrait dépasser le milliard de dollars en 2015. On constate aujourd'hui que l'écosystème d'Arduino par exemple s'est considérablement diffusé et que les cartes d'interface, capteurs, actionneurs et autres accessoires peuvent se trouver sur eBay à des prix extrêmement bas. Par ailleurs, des magasins spécialisés se sont créés et fournissent du matériel et du support sur des forums très actifs (par exemple Sparkfun Electronics). La société Arduino réalise un chiffre d'affaires d'un million de dollars alors qu'elle a ouvert toute sa propriété intellectuelle, tant sur le matériel que sur le logiciel (la seule propriété qu'elle a conservé sont les droits portant sur sa marque), et qu'elle est entourée de concurrents. Finalement les effets économiques produits par l'écosystème qu'elle a initié compensent largement le contrôle qu'elle a abandonné en ouvrant son projet.

Le logiciel libre est un phénomène cumulatif qui permet de produire des infrastructures définitivement disponibles pour la société. Cela ne remet en cause ni la pertinence ni l'intérêt du secteur privé du développement logiciel, bien au contraire : les outils libres pour tous permettent de développer plus vite des applications de meilleure qualité et plus facilement interoperables sans avoir à réécrire sans fin les mêmes fonctions. Dès lors qu'un logiciel libre apporte une réponse à un problème donné, il n'est pas nécessaire de re-développer cette fonction, il suffit de s'en servir. Compte tenu de son caractère non rival, vouloir la re-développer est inutile et contre-productif. Pour reprendre l'analogie évoquée plus haut, l'entrepreneur qui souhaite se lancer dans le transport de fret, qui commencerait à construire des routes entre les sites de ses clients, serait pris pour un déséquilibré. Le recours au réseau routier national – développé sur des fonds publics – lui permet de se concentrer sur son métier de la façon la plus efficace pour lui, à savoir mettre en place une organisation qui lui permettra d'exploiter au mieux sa flotte de véhicules. Cet exemple est transposable au logiciel libre et, de manière plus générale, à tous les biens non rivaux et/ou mutualisables.

4.8 *Open standard*, l'*openness* appliquée à la standardisation

La montée d'Internet a également provoqué d'importantes modifications dans le domaine de la standardisation avec l'arrivée des standards

ouverts notamment grâce à l'IETF (*Internet Engineering Task Force*). Ce groupe de travail qui publiait des spécifications ouvertes concernant les technologies d'Internet a provoqué une petite révolution en ouvrant non seulement les standards mais, plus important encore, l'ensemble du processus de standardisation. Tant que le groupe, créé en 1986, comportait quelques dizaines ou centaines d'adhérents, son impact était resté limité mais il connut une forte montée en puissance au début des années deux mille. La nature du processus de spécification confère une grande robustesse aux standards issus de l'IETF : les spécifications sont élaborées de façon collaborative et publique puis ne sont validées qu'après que deux démonstrateurs développés indépendamment aient été présentés. Le succès de ce modèle l'a rendu contaminant, il a par exemple amené l'IEEE à ouvrir ses propres standards. De même, compte tenu de leur robustesse l'ISO adopte tels quels les standards de l'IETF car il est difficile de faire mieux.

4.9 Les modèles économiques du logiciel libre aujourd'hui ?

L'observation de la situation à l'approche de 2013 permet de pointer quelques tendances. Une première à propos du marché relativement récent des *application stores* (Apple AppStore et Google Play) qui sont installés sur des téléphones ou des tablettes embarquant des systèmes d'exploitation basés sur des noyaux Unix (respectivement BSD et GNU/Linux). Ce modèle, que l'on pourrait qualifier de *Tollware* ou logiciel à péage, s'étend rapidement. Son principe consiste à mettre une plate-forme dotée de fonctionnalités de base en libre accès et de proposer des fonctions ou des contenus supplémentaires prenant la forme de programmes, plugins, etc. Le gros intérêt de cette architecture modulaire est de permettre à des développeurs individuels ou à de très petites entreprises, et donc à une très large population d'acteurs potentiels, de participer activement à l'écosystème. À ce titre Apple, dont l'AppStore a disséminé plus de trente milliards d'Apps, a déjà versé plus de cinq milliards de dollars aux développeurs souvent indépendants¹.

1. Horace DEDIU, « The iOS App market sales run rate is 4 billion per year », dans : (2012), URL : <http://www.asymco.com>.

Cette coexistence de la multinationale et de « l'artisan »¹ montre la grande diversité d'échelle des entités qui peuvent participer à un même écosystème.

Une autre tendance est la multiplication d'écosystèmes fermés (par un péage ou une délimitation organisationnelle), selon un modèle qui s'apparente aux forges logicielles privées. Le principe consiste à mutualiser les développements entre un nombre limité d'acteurs, ce qui induit une limitation proportionnelle de l'effet de levier et de la puissance de débogage.

Les métiers de l'ingénierie des communautés sont en plein développement et leur rôle pour les années à venir s'annonce stratégique pour la plupart des organisations. Face à la pénurie qui se dessine ce métier voit la création des premiers cursus académiques spécialisés, comme c'est déjà le cas dans la Silicon Valley². Il est raisonnable de supposer que la décennie à venir verra une généralisation de ces formations dans les universités européennes.

À cet égard les communautés d'utilisateurs sont en plein essor notamment dans des branches professionnelles. Souvent leur objectif premier est de réaliser des économies en mutualisant les coûts de développement de logiciels métier et de capitaliser en achetant collectivement des logiciels à code ouvert, cependant elles vont souvent plus loin que cela en offrant des services mutualisés (support, formation, etc.).

Le lecteur s'étonnera peut-être que nous n'ayons pas parlé du *desktop* dans ce chapitre. Cette absence est volontaire. L'ordinateur de bureau ne nous paraît pas destiné à un avenir aussi prometteur que des terminaux comme le smartphone, la tablette ou des dispositifs dont le format reste à inventer. Ensuite, la concurrence pourtant vive qui règne entre les différents bureaux des distributions GNU/Linux a échoué à établir un standard stabilisé et il paraît bien tard pour y remédier. L'arrivée de HTML5 pourrait constituer l'amorce d'une solution, sous réserve d'une convergence des différentes versions de cette syntaxe.

1. Maxence GUESDON, *Sur le métier de développeur de logiciels*, 2012, URL : <http://www.oranadoz.net/sur-le-metier-de-developpeur-de-logiciels/>.

2. Par exemple le COSI, Center for Open Source Investigation, fondé par la Carnegie Mellon University que dirige Tony Wasserman.

Conclusion

Le mouvement du logiciel libre est une instance du principe plus général d'*openness*. Il s'agit du modèle sur lequel est construit le modèle scientifique basé sur la production, le partage et la diffusion des connaissances. L'*openness* produit naturellement de l'abondance dès que cela est possible, de la qualité, des communautés fortes et de l'innovation. Ce principe, s'il ne permet pas un retour financier direct produit en revanche une situation propice à une activité économique à travers un écosystème. Dans cette perspective, le logiciel libre permet de générer de la valeur sous la forme de produits reproductibles et améliorables sans limite dont la mise en œuvre crée puis entretient un écosystème. Ce dernier offre de nouvelles potentialités de marchés et donc d'activités économiques permettant de récupérer, indirectement, la valeur investie initialement, et parfois bien au-delà. On constate donc que l'*openness* en général, et le logiciel libre en particulier, n'est pas un modèle économique en soi mais un facilitateur de modèles économiques à travers l'écosystème qu'il produit. Les domaines les plus susceptibles de produire des succès sont ceux qui permettent de générer l'effet de levier le plus important, autrement dit ceux qui concernent les problèmes les plus communs et partagés par le plus grand nombre de personnes ou d'organisations, comme les biens non rivaux. Pour autant, comme nous l'avons vu avec l'*open hardware*, les produits rivaux, bénéficient aussi grandement de l'*openness* qui s'applique alors à leur environnement informationnel et cognitif.

Sur l'axe allant des infrastructures aux applicatifs, le type de demande remplissant le mieux ces critères est le besoin d'infrastructures et ce qui y répond le moins bien sont les applicatifs très spécifiques dont on comprend qu'ils n'offrent pas de grands effets de levier. Cela permet de tracer un modèle hiérarchisé s'étendant du niveau « infrastructures globales » au niveau « applications spécifiques ».

En somme, pour fonctionner correctement les modèles économiques du logiciel libre doivent chercher un effet indirect car la licence d'un logiciel – véritablement – libre étant par définition ouverte et gratuite, il faut gagner sa vie « autour » et pas « dessus ». Par ailleurs les modèles pseudo-libres, c'est-à-dire ceux qui tentent d'allier les avantages du libre et du propriétaire, n'atteignent pas une diffusion suffisante pour établir un éco-

système et ne parviennent donc pas à générer une activité économique viable.

Les grandes organisations paraissent commencer à se tourner vers le logiciel libre : la santé, l'industrie automobile, le gouvernement. De même, des domaines applicatifs comme la gestion de grands ensembles de données seront probablement de plus en plus utilisateurs, et souhaitons-le contributeurs, au logiciel libre, notamment à travers les données ouvertes, *open data*, dont le potentiel est aujourd'hui notoirement sous-estimé – et sous-exploité – en Europe (contrairement aux États-Unis et à l'Asie).

Ce chapitre se refermera en mentionnant quelques défis que nous devons relever dans le mouvement du logiciel libre.

Il nous faut, tout d'abord, parvenir à séduire les personnes n'ayant pas d'aptitudes techniques dans le domaine de l'informatique. Elles sont très nombreuses, beaucoup plus que celles qui en disposent. Le gisement d'écosystèmes potentiels est considérable. Il nous faut pour cela convaincre les concepteurs, les ergonomes, les cognitivistes, les sociologues, anthropologues, ethnologues, que le logiciel libre n'est pas le domaine réservé des informaticiens, il peut et doit aussi devenir le projet de chacun. Nous devons aussi expliquer ou réexpliquer à nos politiques, à nos dirigeants, à nos industriels, ce que leurs prédécesseurs du siècle des Lumières avaient fort bien compris : l'*openness* est la solution la plus efficace que l'on connaisse pour créer de l'innovation et de la prospérité, elle doit être la priorité numéro un de toute politique visant un développement économique. Souvenons-nous également que la fonction première et essentielle du secteur public est de préparer le terreau infra-structurel sur lequel le secteur privé et l'économie pourront prospérer.

Pour pouvoir relever ces défis, il est indispensable d'informer, de sensibiliser et de former tous ceux qui le souhaitent à l'*openness*, c'est-à-dire a) à la gouvernance de communautés, b) aux problématiques liées aux licences et c) au développement collaboratif de logiciels informatiques. Il est tout aussi indispensable d'ouvrir l'information et la connaissance et de renvoyer les inventions, notamment celles financées par le secteur public, vers le tissu social afin de faire émerger des écosystèmes qui, nous l'avons vu, constituent la condition préalable de tout développement économique.

Agir selon la philosophie de l'*openness* n'est pas une utopie, bien au contraire les faits indiquent qu'il s'agit d'une attitude extrêmement

pragmatique produisant, indirectement, des effets économiques majeurs. L'illustration des soieries lyonnaises est un cas parmi d'autres et l'Histoire montre de manière récurrente que si le mouvement d'adoption de l'*openness* est suffisamment ample, il est susceptible produire un impact sociétal de forte magnitude menant à de longues périodes de prospérité économique.

Voilà de beaux défis qui nous attendent !

Références

- BERNERS-LEE, Tim, *The writing code, Stata Center Opening Symposium*, 2007, URL : <http://www.thewritingcode.com/pdfs/TimBLTranscript.pdf>.
- BORDAGE, Frédéric, « Open source : la naissance d'un écosystème », dans : *01Net* (2005), URL : <http://pro.01net.com/editorial/270447/open-source-la-naissance-dun-ecosysteme/>.
- DAVID, Paul A., *An Essay on Patronage, Reputation and Common Agency Contracting in The Scientific Revolution*, Stanford : Stanford University, 2007.
- DEDIU, Horace, « The iOS App market sales run rate is 4 billion per year », dans : (2012), URL : <http://www.asymco.com>.
- ÉLIE, François, « Mobile, Cloud and the Digital Native Generation », dans : *Conference fOSSa*, INSA Lyon, 2011.
- FORAY, Dominique et Liliane Hilaire PEREZ, « The economics of open technology : collective organization and individual claims in the "fabrique lyonnaise" during the old regime », dans : *Frontiers in the economics of innovation and new technology. Essays in honor of Paul A. David*, sous la dir. de Cristiano ANTONELLI et al., Cheltenham : Edward Elgar Pub., 2005, p. 239–254.
- GARDLER, Ross, « Community over code », dans : *Conference fOSSa*, Grenoble, 2010.
- GUESDON, Maxence, *Sur le métier de développeur de logiciels*, 2012, URL : <http://www.oranadoz.net/sur-le-metier-de-developpeur-de-logiciels/>.
- MOGLEN, Eben, *Innovation under Austerity*, F2C Freedom to Connect, 2012.

RAYMOND, Eric S., *The Cathedral and the Bazaar*, Sebastopol, CA :

O'Reilly, 2001, URL : <http://www.catb.org/esr/writings/homesteading/>.

SCOFFONI, Philippe, *LibreOffice, OpenOffice un fork vu de l'intérieur*,

2011, URL : <http://philippe.scoffoni.net/libreoffice-openoffice-fork-vu-de-linterieur>.

TIEMANN, Michael, *History of the OSI*, 2012, URL : [http://www.](http://www.opensource.org/history)

[opensource.org/history](http://www.opensource.org/history).

WENGER, Etienne, *Communities of Practice : Learning, Meaning, and*

Identity, 1^{re} éd., Cambridge, Mass. : Cambridge University Press, 1999.

Le logiciel libre : un renouveau du modèle industriel coopératif de l'informatique ¹

Nicolas JULLIEN
Jean-Benoît ZIMMERMANN

Cela fait maintenant plusieurs années que le phénomène du logiciel libre a émergé dans la sphère économique et de nombreux travaux scientifiques s'y sont intéressés, parmi lesquels ceux publiés dans *Terminal* en 1999² qui ont été précurseurs. Un logiciel libre est un logiciel dont le code source est rendu librement accessible de manière à pouvoir être utilisé, modifié ou intégré dans d'autres logiciels dans des conditions qui sont définies par le type de licence qui lui est attachée et qui constitue une forme de contrat entre son ou ses auteurs et ses futurs utilisateurs. Aujourd'hui, le logiciel libre semble devenu un enjeu économique majeur, notamment pour l'Europe, comme en témoigne un rapport déjà ancien publié par la Commis-

1. Cet article se base sur des travaux des auteurs, notamment dans le cadre du projet européen Calibre et dans le cadre du projet ANR CCCP-Prosodie (<http://cccp-prosodie.org>). Cf une version complète de cet article sur <http://ssrn.com/abstract=1360683>.

2. Voir « Logiciels Libres : de l'utopie au marché », *Terminal*, numéro spécial (80-81), Automne-Hiver 1999.

sion Européenne fin 2006¹, ou l'analyse de Dang Nguyen & Genthon² : « en focalisant sur un programme ambitieux de production de logiciel libre dans les systèmes embarqués et les réseaux domestiques, l'Europe peut atteindre plusieurs objectifs : permettre le libre accès à une ressource essentielle, stimuler la concurrence, faciliter la réalisation des objectifs de Lisbonne, et restaurer la compétitivité européenne dans les TIC. »

S'il s'agit toujours d'une organisation non marchande de production, ses produits intéressent le monde marchand, depuis IBM³ qui a annoncé en 2001 investir plus d'un milliard de dollars dans Linux jusqu'à la *shared source initiative* de Microsoft⁴, qui reprend l'idée du partage du code source avec les utilisateurs, et distribue même certains de ses logiciels sous une licence *open source*⁵, en passant par l'État français, qui a reconnu l'excellence française dans ce domaine, et autorisé la création d'un « pôle » logiciel libre au pôle de compétitivité Systematic⁶, ou encore l'entreprise Red Hat qui a récemment franchi le cap du milliard de dollars de chiffre d'affaires⁷.

Le phénomène ne concerne plus seulement les informaticiens ou le marché des grandes organisations (« grands comptes informatiques ») : les entreprises proposant des solutions *libres* se tournent aussi de plus en plus vers les PME, comme en témoignait déjà le dossier du *Journal du Net* de septembre 2005⁸. Lakhani et Wolf⁹ relevaient que, parmi les développeurs de logiciel libre ayant répondu à leur enquête, une majorité des

1. Voir « le logiciel libre est bon pour l'économie européenne, l'emploi des informaticiens et la compétitivité des entreprises... », URL : <http://ec.europa.eu/enterprise/ict/policy/doc/2006-11-20-flossimpact.pdf>.

2. Godefroy DANG NGUYEN et Christian GENTHON, « Les perspectives du secteur des TIC en Europe », dans : *Cahiers de Recherche Marsouin* 4 (2006), URL : <http://www.marsouin.org>.

3. La page d'IBM retraçant la « saga Linux » : <http://www-5.ibm.com/e-business/ch/fr/linux/growing.html>.

4. <http://www.microsoft.com/resources/sharedsource/default.mspx>.

5. http://solutions.journaldunet.com/0404/040407_microsoft.shtml.

6. <http://www.systematic-paris-region.org/fr/logiciel-libre>.

7. Steven J. VAUGHAN-NICHOLS, « Red Hat : The first billion dollar Linux company has arrived », *www.zdnet.com*, 2012.

8. <http://solutions.journaldunet.com/dossiers/libre/sommaire.shtml>.

9. Karim R. LAKHANI et Robert G. WOLF, « Why Hackers Do What They Do : Understanding Motivation and Effort in Free/Open Source Software Projects », dans : *Perspectives on Free and Open Source Software*, sous la dir. de Joseph FELLER et al., Cambridge, Mass. : The MIT Press, 2007, p. 3–22.

répondants étaient des professionnels de l'informatique et environ 40% étaient payés pour participer à des projets libres/*open source*. Enfin, l'intérêt du monde marchand pour le modèle du libre dépasse le cadre de l'informatique pour se diffuser aussi dans les télécommunications ¹.

Tous ces éléments traduisent une situation qui peut être entendue sous l'angle de l'émergence d'un marché, au sens d'un échange marchand de biens et de services et être analysée dans une perspective d'économie industrielle. En même temps, la production de logiciel libre repose d'abord sur une collaboration, une co-production, sans échange financier, entre différents producteurs. C'est cette articulation entre une production collective, non marchande, que nous qualifierons dans la suite de « coopérative », et son succès commercial qui fait toute la richesse de l'écosystème que nous nommerons « le Libre », et que nous nous proposons d'expliquer ici.

D'un point de vue général, nous nommerons « libre » toute création de nature intellectuelle (artistique, littéraire, scientifique et plus généralement de connaissance) dont l'accès et l'usage sont autorisés sans exclusivité, sous la seule condition de ne pas en altérer les conditions. En ce sens la notion de Libre s'apparente à la notion économique de « commun » qui désigne le partage d'une ressource ou d'un ensemble de ressources que celles-ci aient une origine naturelle, de production privée ou de production publique. On comprend facilement que la notion de commun est inséparable des modalités de gouvernance qui lui sont attachées en vue de définir les conditions de son usage, de sa préservation et de sa reproduction, voire de son aliénation. On peut par exemple penser aux conditions d'exploitation d'une ressource de pêche. En ce qui concerne le logiciel libre, cette définition permet de comprendre pourquoi la question du droit d'auteur reste au cœur de la problématique, au contraire du domaine public caractérisé par une absence ou une disparition de toute propriété intellectuelle.

Dans une première partie, nous expliquerons comment le modèle de production coopératif trouve ses racines très loin dans l'histoire de l'industrie informatique puisqu'il était celui des origines ; pourquoi il a été

1. Dans le domaine de la voix sur IP, le logiciel libre Asterisk, <http://www.asterisk.org/>, est une des offres les plus utilisées. Motorola a choisi un système Linux pour l'un des trois systèmes d'exploitation de ses terminaux mobiles.

mis à mal par la marchandisation du logiciel et le développement de stratégies propriétaires et d'incompatibilité ; et comment il renaît dans les années 1980 et surtout 1990, porté par le formidable outil de diffusion et d'échange que constitue Internet (là aussi, et dans la suite de l'article, en tant qu'écosystème, basé sur un protocole réseau, des applications et des usages). Mais il y a lieu de s'interroger sur la durabilité du modèle confronté une fois encore aux impératifs de la sphère marchande. C'est pourquoi la deuxième partie de ce chapitre s'applique à dégager les avantages comparatifs du mode de production libre par rapport à celui dit propriétaire, qui pourraient fonder sa pérennité dans un contexte profondément transformé des modes de production, de distribution et d'utilisation en ce début de troisième millénaire.

1. Le modèle coopératif de production du logiciel, des origines au renouveau porté par Internet

1.1 Aux origines, une production coopérative de logiciel

Au début de l'informatique, dans les années 1950, la production d'ordinateurs était presque exclusivement une production américaine. L'activité informatique était très liée aux projets militaires, aux financements militaires et aux projets de recherche : l'État américain a financé la recherche et le développement de l'informatique à ses débuts et la plupart des machines produites à cette époque l'ont été pour ce « client »¹.

Les machines étaient livrées « nues » par le fournisseur, c'est-à-dire sans programmes. Le fournisseur aidait son client à mettre en œuvre ses machines et à développer les programmes qui lui permettaient de les utiliser. Le logiciel était alors considéré comme un outil et un objet de recherche. Sa production reposait sur des relations de collaboration poussées entre centres de recherche publics et industrie² et les principaux utilisateurs

1. La part du financement étatique de la recherche des entreprises d'informatique américaines entre 1949 et 1959 est estimée à 59%, en moyenne pondérée. Voir Christian GENTHON, *Croissance et crise de l'industrie informatique mondiale*, Paris : Syros, 1995, p. 33. Cette part va de 46% pour l'entreprise RCA à 90% pour Raytheon. C. Genthon estime que le parc public, y compris le parc militaire, représentait encore, en 1962, 40% du parc installé en valeur.

2. Nathan ROSENBERG, *Inside the Black Box : Technology and Economics*, Cambridge, MA : Cambridge University Press, 1983.

de logiciel étaient alors les principaux producteurs (les universités, les grandes entreprises, surtout celles liées à l'industrie de défense et, bien sûr, le principal producteur de machines informatiques, IBM).

L'ensemble des logiciels produits était disponible, sans droits de propriété ; la notion même de propriété intellectuelle du logiciel n'existait pas. La seule façon de protéger un logiciel était de le garder secret, ce qui n'était pas la stratégie soutenue par les institutions publiques et par les entreprises informatiques émergentes. En effet, la diffusion de logiciels dans le domaine public était directement bénéfique aux producteurs : plus la quantité de logiciels disponibles était grande, plus l'étaient les incitations à acheter un ordinateur. De plus, le département américain de la défense (DoD) a encouragé ces pratiques en finançant des programmes de recherche et de développement coopératifs universités-entreprises. Il a aussi favorisé la diffusion des « bonnes pratiques » de programmation¹.

On le voit, la production coopérative n'est pas neuve. Elle a même dominé tous les débuts de l'informatique. Pour comprendre comment les choses ont évolué, il est nécessaire de prendre en compte les stratégies industrielles des grandes firmes de l'informatique et tout particulièrement celle de leur leader, IBM. En effet, au-delà de la demande publique originelle, ces entreprises entendaient valoriser leur activité sur un vrai marché, celui des « grands comptes », grandes entreprises et grandes administrations, principalement en Amérique et en Europe, et bénéficier ainsi d'importants effets d'échelle². IBM en particulier menait une politique commerciale très offensive chez ses grands clients. S'appuyant sur les « directions informatiques », états dans l'État dont elle avait soutenu la création au sein des structures utilisatrices, elle y vendait des solutions à des problèmes, principalement de l'ordre de la gestion et de l'administration, c'est-à-dire des combinaisons appropriées de matériels et de logiciels, en réponse aux besoins du client et à ses caractéristiques propres. Cette manière de faire permettait ensuite de fonder des arguments de vente

1. « The US armed forces, from the earliest days of their support for the development of computer technology, were surprisingly eager for technical information about this innovation to reach the widest possible audience ». R. N. LANGLOIS et D. C. MOWERY, « The Federal Government Role in the Development of the U.S. Software Industry : An Assessment », dans : *The International Computer Software Industry : A Comparative Study of Industry Evolution and Structure*, sous la dir. de D. C. MOWERY, Oxford : Oxford University Press, 1996, p. 58.

2. Les effets d'échelle désignent les rendements croissants de production, c'est-à-dire la diminution des coûts unitaires de production avec le nombre d'unités produites.

de nouveaux éléments coûteux, notamment des périphériques, afin d'inciter le client utilisateur à une utilisation plus efficace des capacités de son système. Or, c'est précisément ce type de stratégie commerciale qui a été visé par la législation anti-trust américaine en 1969, qui voyait dans cette pratique commerciale, un abus de position dominante, fondé sur la facturation jointe du logiciel et du matériel. Injonction a donc été faite aux producteurs informatiques de procéder à la facturation séparée (*unbundling*) du logiciel et du ou des matériels. Le logiciel acquerrait ainsi un statut de marchandise et, en tant que tel, requerrait d'être protégé (ce qui fut fait par recours au droit d'auteur).

Puisqu'il devait être facturé séparément, le logiciel devenait alors l'objet de transactions marchandes et donc d'un marché, ce qui apparut rapidement contradictoire avec la libre circulation et la production coopérative dont il était l'objet jusque-là. Ainsi, le paradoxe est que, de par les lois du marché, cette production coopérative s'est marginalisée alors qu'elle semblait satisfaire l'ensemble des acteurs, producteurs et utilisateurs de logiciels. Un certain nombre d'acteurs ont su tirer profit de cette situation nouvelle.

1.2 Une production coopérative de plus en plus marginale, même si elle reste importante pour l'innovation en informatique

Le marché des logiciels et des services se développe vraiment à partir des années 1970. Le système de production coopératif et hors marchand est alors petit à petit occulté. Il ne disparaît pas complètement pour autant, ainsi que le montre l'exemple du système d'exploitation pour station de travail et serveur Unix. D'abord développé de façon collaborative par des entreprises et des universités, Unix était inspiré du projet de recherche « Multics », qui regroupait les Bell Labs d'AT&T, le MIT et General Electric, avec des financements fédéraux. Faute de ressources, il est abandonné en 1968. Il est repris par deux informaticiens des Bell Labs, Dennis Ritchie et Ken Thompson, rejoints en 1970 par Brian Kernighan. La première version documentée d'Unix fut terminée en 1971 et diffusée librement (AT&T, du fait de son monopole dans les télécommunications, n'avait pas le droit d'avoir des activités commerciales en informatique). Elle s'impose rapidement dans les universités d'informatique, qui contri-

buent fortement à son évolution (principalement l'université de Berkeley, à partir de 1977)¹.

Mais il y avait une séparation de plus en plus nette entre le système de production, coopératif, souvent issu du milieu de la recherche, qui pouvait initier des projets, et le système qui distribuait et adaptait le projet aux besoins des utilisateurs. Pour le dire autrement, des entreprises se sont approprié ces logiciels, et en ont assuré la distribution et l'évolution.

Encore une fois, il est nécessaire de prendre en compte les stratégies industrielles des constructeurs informatiques pour comprendre la situation. La stratégie commerciale d'IBM et des grands constructeurs d'« ordinateurs universels » (*mainframes*), consistant à vendre des « solutions » clefs en main aux grands comptes, avait été ébranlée par la législation anti-trust. Parallèlement, les progrès technologiques, surtout liés à la miniaturisation croissante des composants et à la généralisation des circuits intégrés avaient permis de concevoir des ordinateurs plus petits et moins coûteux, permettant une ouverture du marché vers des clients de moindre taille. Un nouveau venu, DEC (Digital Equipment Corporation), en lien étroit avec le MIT, s'était engouffré dans la brèche en introduisant un nouveau concept commercial, celui des mini-ordinateurs, systèmes de petite taille commercialisés nus, dotés de leur seul système d'exploitation.

Il revenait alors à l'utilisateur de spécifier ses propres besoins d'utilisation et de développer ou de faire développer les logiciels d'application correspondants. Cette nouvelle approche consacra la séparation entre les activités de production d'ordinateurs et celle de développement de logiciels d'utilisation. Les constructeurs d'ordinateurs restèrent bien sûr concernés par les systèmes d'exploitation que DEC s'évertuait à rendre compatibles au sein de son offre d'ordinateurs tandis qu'IBM butait sur ses propres incompatibilités y compris au sein d'une même gamme de machines.

Mais le plus important pour notre propos était que les logiciels d'application, en tant que tels, et non plus incorporés dans une machine, étaient désormais l'objet d'une demande à laquelle répondaient un nombre croissant d'entreprises spécialisées, les *software houses*. Un véritable marché

1. Voir Marshall K. MCKUSICK, « Twenty Years of Berkeley Unix. From AT&T-Owned to Freely Redistributable », dans : *Open Sources : Voices from the Open Source Revolution*, sous la dir. de Chris DiBONA, Sam OCKMAN et Marc STONE, Sebastopol, CA : O'Reilly, 1999.

du logiciel est né qui restera longtemps encore dominé par une offre sur mesure, proche du service, en attendant que de véritables espaces de compatibilité se fassent jour dans les années 1980, permettant la portabilité de logiciels standards.

Dans le cas d'Unix, AT&T qui fut autorisée à entrer sur le marché de l'informatique en 1984 a alors demandé le paiement de ses droits sur le nom Unix et sur le système dont elle était à l'origine. Dès le début des années 1980, l'entreprise SUN utilisait, pour faire fonctionner ses stations de travail, un système d'exploitation qu'elle avait adapté d'une version issue de Berkeley, système dont elle se réservait les possibilités d'évolution. Dans son sillage, IBM, HP, DEC et bien d'autres se sont rangées à cette logique propriétaire en offrant chacune à leurs clients leur propre version d'Unix, incompatible avec celle de ses concurrents. Plus près de nous, dans les années 2000, Apple s'est approprié la version 5 de FreeBSD (un descendant libre de l'Unix de Berkeley) pour en faire le système d'exploitation MacOS X. Ce nouveau partitionnement d'un monde qui aurait pourtant pu s'unifier est source d'inefficacité économique (coûts de développement, coûts de migration, moindre croissance du marché...).

Pour tenter d'y remédier, un mouvement a vu le jour autour de la notion de systèmes ouverts. En particulier est né en 1984 le groupe X/Open fondé par un ensemble international de grands constructeurs d'ordinateurs. Son objectif était de « faciliter et orienter la démarche des utilisateurs de l'administration et du secteur privé, des vendeurs de logiciels, des organismes de normalisation et des fabricants de systèmes anxieux de résoudre les dilemmes technologiques résultant de l'incompatibilité de leurs divers systèmes et logiciels. »¹

Si le système de production coopératif était toujours à la base du développement de nombreux logiciels, notamment dans la recherche publique, le lien entre la demande (les utilisateurs) et les producteurs n'était désormais plus direct, mais passait par l'intermédiaire des entreprises. Il y avait d'un côté la recherche, coopérative, productrice de prototypes et financée par des subventions ou des commandes publiques et, de l'autre, l'industrie qui s'appropriait ces prototypes et qui les distribuait.

1. Pamela A. GRAY, *Open systems : a business strategy for the 1990's*, London : McGraw-Hill, 1991.

Cette tendance s'affirma dans les années 1990 : ces années correspondent à un relatif désengagement des institutions fédérales américaines du financement des projets logiciels et donc à une plus grande difficulté pour initier et financer de tels projets. De plus, la recherche prit une orientation plus théorique (algorithmique, tests de stabilité des programmes, etc.). Il semble que le modèle coopératif de production du logiciel se soit transformé pour, finalement, aboutir à un schéma plus classique : les centres de recherche publics inventent de nouveaux outils et de nouveaux concepts, publiés sous des licences permettant aux industriels de se les approprier et de les transformer en objets utilisables commercialement.

Pourtant, les entreprises ont continué à s'alimenter à la source des progrès techniques que représentent les centres de recherche académiques et ont bénéficié de l'émergence de nouveaux concepts comme les langages orientés objet ou les nouvelles architectures client-serveur qu'elles ont rapidement adoptés et se sont appropriés. D'autre part, la collaboration dans la recherche a été facilitée, par la diffusion de réseaux connectant les universités. Notons qu'avant même la naissance d'Internet, dans les années 1970, les chercheurs américains ont bénéficié du réseau ARPANET, ancêtre d'Internet, mis en place et financé par le département de la défense des États-Unis pour mailler le territoire américain à des fins de sécurité nationale et pour favoriser les échanges d'idées et d'outils entre universités et industries dans le domaine des technologies de l'information et de la communication. D'autres chercheurs, ailleurs, bénéficiaient des réseaux expérimentaux fondés sur la commutation par paquets et la mise au point d'architectures d'interconnexion de systèmes hétérogènes, comme le réseau français Cyclades promu par l'INRIA ¹.

C'est la diffusion d'Internet et des outils Web, associée à une innovation juridique, la licence GPL, mais aussi les limites du système fermé, propriétaire, qui expliquent le renouveau de la production coopérative.

1. Le réseau Cyclades peut lui-même être considéré parmi les projets qui ont donné naissance à Internet, mais aussi aux architectures d'interconnexion OSIA recommandées par l'ISO. Voir [http://fr.wikipedia.org/wiki/Cyclades_\(réseau\)](http://fr.wikipedia.org/wiki/Cyclades_(réseau)). Louis POUZIN, « The Cyclades computer network », dans : *ICCC 2* (1982).

1.3 Un renouveau de la production coopérative grâce à Internet : épiphénomène ou tendance de fond ?

Les protocoles à la base d'Internet (comme TCP/IP), et les logiciels qui les implémentent, mais aussi les systèmes d'exploitation libres comme GNU/Linux, FreeBSD ou OpenBSD ont pour la plupart été développés dans les universités et les centres de recherche publics¹.

C'est au CERN qu'a été développé le langage HTML et son système de liens hypertextes. Aujourd'hui encore, la grande majorité des logiciels d'Internet sont des logiciels libres². Et il n'est pas un hasard que ce soit dans le domaine des serveurs web que les logiciels libres occupent la position la plus importante avec 65% du marché détenus par Apache³. Ici plus qu'ailleurs ce marché a bénéficié d'une proximité utilisateurs-développeurs sur laquelle s'est bâtie l'alternative *open source*.

Les logiciels libres ont commencé à être connus en dehors de cette sphère spécialisée parce que des utilisateurs se sont aperçus qu'ils répondaient à leur demande de nouveaux services, autour d'Internet et des réseaux informatiques : serveurs de fichiers, serveurs d'imprimante. Le succès des logiciels libres repose donc d'abord sur une adéquation historique à de nouveaux marchés. Ils ne se sont pas imposés face à d'autres logiciels pour répondre à de nouveaux besoins mais se sont diffusés en même temps que les besoins auxquels ils répondaient.

Internet constitue aussi un formidable outil de diffusion. Très tôt, il a été utilisé par les producteurs de logiciels libres, qui y avaient accès dans leurs universités ou dans leurs centres de recherche. Ils ont pu tirer parti de l'ensemble des nouveaux supports de diffusion rendu ainsi disponibles (listes de diffusions, news, sites Web) pour faire connaître ces logiciels et les rendre plus accessibles à de nouveaux utilisateurs.

Mais la diffusion d'Internet n'a pas seulement permis la diffusion des logiciels existants. Elle a aussi rendu plus facile leur production par la mise en relation, des développeurs et donc l'organisation de projets connectant

1. Les Unix libres BSD ont été développés à partir de l'Unix de Berkeley (BSD signifie « Berkeley Software Distribution »).

2. Voir les chiffres de Netcraft sur la diffusion des logiciels serveurs Internet, <http://news.netcraft.com/archives/category/web-server-survey>.

3. <http://www.journaldunet.com/developpeur/outils/part-de-marche-des-serveurs-web>.

des communautés de contributeurs potentiels via des espaces d'échanges, de discussion, d'interactions, de production et d'évaluation.

Comme nous l'avons dit plus haut, les réseaux, et notamment les réseaux USENET existaient avant Internet, ce sont eux qui ont permis d'initier les projets libres et de faire se rencontrer des développeurs qui avaient des besoins proches. Cependant, les débits de ces réseaux étaient faibles et les services assez pauvres. Il est certain que la diffusion d'Internet hors de la sphère de la recherche a permis aux développeurs de disposer d'un outil de coordination plus efficace : c'était le premier système qui leur offrait un accès simple à l'information et aux systèmes de stockage, depuis n'importe quel point d'entrée (et notamment de leur domicile), pour un coût modique. Internet a aussi permis la multiplication de forums spécialisés et a augmenté la probabilité que des utilisateurs-développeurs ayant des besoins spécifiques proches se rencontrent et atteignent la « masse critique » qui permet d'initier un projet.

Le renouveau de la production coopérative est donc dû non seulement à l'adoption des logiciels produits par une grande population d'utilisateurs (que nous avons qualifié d'adéquation historique à de nouveaux marchés), entraînant un regain du volume de logiciels produits de façon coopérative (nous renvoyons aux chiffres de l'introduction), développant ainsi la population potentielle d'utilisateurs, etc.

Qu'en est-il toutefois du devenir d'un tel mode de production confronté aux impératifs de la sphère marchande ?

La première vague d'utilisation industrielle des logiciels libres, autour et grâce à Internet, s'explique par des stratégies que l'on peut qualifier d'opportunistes. Les motivations qui conduisaient les entreprises de l'informatique à utiliser et contribuer au développement de logiciels libres étaient principalement indirectes et résultaient notamment de la concurrence qu'elles se livraient pour le contrôle des standards. D'un côté, les stratégies mises en œuvres par certaines entreprises visaient à affaiblir leur(s) concurrent(s) direct(s) en subventionnant un standard antagoniste à celui sur lequel ce ou ces concurrents appuyaient leur offre (ex. : IBM ou HP subventionnant Linux pour concurrencer SUN, dominant sur le marché Unix). Mais, de l'autre, les logiciels fonctionnant rarement de façon isolée, il fallait assurer leur cohérence, leur « mise en compatibilité »

(phénomène d'interrelation technologique) avec les logiciels (libres) standards d'Internet. Les entreprises qui dépendaient d'un standard pour réaliser leur activité (comme Sun, producteur de Java avec le serveur Web Apache) avaient, de plus, tout intérêt à ce que ce standard soit le plus ouvert possible, afin ne pas dépendre de la stratégie de l'entreprise qui le fournissait. Elles rejoignaient ainsi la logique d'autres entreprises qui ouvrent très volontairement les standards dont elles sont porteuses afin d'en favoriser l'adoption, comme l'avait fait Adobe concernant son standard d'impression PostScript.

De telles motivations, classiques en informatique, peuvent expliquer le ralliement d'entreprises commerciales au monde du libre. Mais comme l'ont fait remarquer Genthon & Phan¹, le risque est grand que la participation de ces entreprises soit limitée au temps où elles trouvent un intérêt stratégique à affaiblir leurs concurrents.

Le phénomène Unix va-t-il se reproduire ? C'est la thèse que défendaient Genthon & Phan², et c'est ce que craignait Horn³. Ils expliquaient que les entreprises risquaient de se réapproprier la production de ces logiciels soit en les fermant, comme l'a fait Apple, soit en publiant des versions de moins en moins compatibles de certains logiciels phares, comme peuvent le faire Red Hat ou, à l'époque, SuSE avec GNU/Linux⁴. D'autre part, comme l'explique Fitzgerald⁵, l'implication des entreprises dans les communautés du Libre peut changer leur objet, leur mode de fonctionnement et finalement décourager les participants volontaires.

Il faut donc affiner l'analyse, en se demandant en quoi le Libre répond à des besoins non satisfaits par le système de production actuel et notamment pourquoi l'organisation coopérative pourrait être à nouveau la plus adaptée pour répondre aux besoins des utilisateurs et des industriels de l'informatique. Et pour cela, il est utile de comprendre pourquoi et com-

1. Christian GENTHON et Denis PHAN, « Les logiciels libres : un nouveau modèle ? », dans : *Terminal* 80 (1999), p. 167–188.

2. *ibid.*

3. François HORN, *L'économie des logiciels*, Paris : La Découverte, 2004.

4. Il est intéressant de noter que le projet GNU signifie « GNU's Not Unix », mettant en garde, dans l'esprit de son fondateur Richard Stallman, contre une dérive telle que celle qu'avait connue Unix.

5. Brian FITZGERALD, « The transformation of open source software », dans : *MIS Quarterly* 30.3 (2006), p. 587–598, URL : <http://ulir.ul.ie/handle/10344/92>.

ment les spécificités du mode de production du Libre sont susceptibles de fonder des avantages comparatifs sur le modèle dit propriétaire.

2. Les modèles industriels du Libre

Au cours des années 1990, avec l'arrivée de l'Internet, la principale évolution technique en informatique était, bien sûr, la généralisation des réseaux informatiques, tant à l'intérieur des organisations (entendues ici comme les entreprises ou les institutions publiques) qu'entre ces organisations. La miniaturisation a aussi permis l'apparition d'une nouvelle gamme d'objets dits « nomades » (lecteurs de musique, téléphones mobiles, notamment smartphones). Ainsi, en un demi-siècle, on est passé d'un seul appareil, dédié à une tâche déterminée à l'avance et commune à l'ensemble de l'organisation, à plusieurs machines par individu, connectées entre elles et utilisées pour des tâches différentes, évolutives dans le temps. Mise en réseau, échange entre des systèmes hétérogènes, communication entre ces outils sont devenus cruciaux.

Ainsi, la croissance de l'Internet, et la croissance des échanges extérieurs aux organisations ont fait des externalités de réseau le fondement des rendements croissants d'adoption. En d'autres termes chaque utilisateur bénéficie d'avantages liés au fait que d'autres utilisateurs ont déjà adopté les mêmes technologies. Les bénéfices de l'adoption d'une technologie sont donc croissants au fur et à mesure de son adoption par de nouveaux utilisateurs.

Au sein de ces organisations utilisatrices, la demande est devenue de plus en plus hétérogène avec la mise en réseau des différents systèmes et la nécessité pour les utilisateurs travaillant dans l'entreprise de partager les mêmes outils. Les logiciels (et plus particulièrement les progiciels) doivent être adaptés aux besoins et aux connaissances de chaque individu, sans perdre l'avantage des économies d'échelle, donc de la normalisation des programmes sur lesquels la solution est fondée. Il devient alors logique que les entreprises utilisatrices recherchent des solutions plus ouvertes qui leur garantissent un plus grand contrôle dans la mesure où elles ne doivent pas faire face à une fragmentation des logiciels utilisés par leurs membres mais peuvent au contraire les intégrer dans un vaste ensemble. Ici, l'atout d'Internet n'est pas tant d'offrir un « protocole » pour

permettre la transmission de données, car il en existait déjà, mais d'en offrir un suffisamment simple et souple qui puisse s'imposer comme un standard pour l'échange.

En parallèle de cette évolution, les technologies de programmation ont évolué¹ : l'arrivée des langages de programmation orientés objet (C++, Java) a permis à des composants logiciels déjà développés d'être réutilisés. De là est née la notion de « logiciels modulaires » : l'idée est de développer un ensemble de petits programmes logiciels (modules ou composants logiciels), qui ont chacun une fonction spécifique. Ils peuvent être associés et utilisés sur n'importe quelle machine grâce à leurs interfaces de communication normalisées. Ce qui caractérise l'évolution technologique des logiciels est donc l'interdépendance croissante entre les logiciels, tandis que les composants logiciels qui sont ré-utilisés sont de plus en plus raffinés et spécialisés². Ce système ne peut fonctionner que si les composants sont en effet réutilisables, c'est-à-dire si les producteurs s'accordent sur un mécanisme qui permette de normaliser les interfaces et de garantir, dans le temps, la stabilité de ces normes.

Horn³ défend l'idée que nous sommes entrés dans un nouveau mode de production : le « sur-mesure de masse », qui s'accompagne d'une augmentation de la part du service dans la vente de logiciels. La transformation d'IBM en une entreprise de « solution », ou de service, ou l'émergence de logiciels paramétrables comme les ERP en sont des exemples. Si l'on considère le passé, ces évolutions de la demande devraient, à leur tour, engendrer une évolution des modèles d'affaire et des modèles de production. Et en regard des évolutions de la demande, l'offre libre présente des avantages importants.

2.1 Des avantages spécifiques au logiciel libre

Les avantages du logiciel libre sont particulièrement visibles dans la gestion de l'amélioration de la qualité et dans le respect des normes.

1. HORN, *op. cit.*

2. Jean-Benoît ZIMMERMANN, « L'industrie du logiciel, de la protection à la normalisation », dans : *Changement institutionnel et changement technologique : évaluation, droits de propriété intellectuelle, système national d'innovation*, sous la dir. de Baslé DUFOUR, CNRS Editions, Paris, 1995, p. 181–207.

3. HORN, *op. cit.*

2.1.1 La qualité du logiciel

Plus que de simples produits de la recherche publique, les programmes libres étaient, d'abord et avant tout, des outils développés par des utilisateurs-experts, au sens que les utilisateurs et les développeurs ne formaient qu'une seule catégorie d'acteurs, pour répondre à leurs propres besoins. Ainsi, la faible qualité des logiciels fermés et, surtout, la difficulté de les faire évoluer sont les raisons fondamentales à l'origine de l'initiative de Richard Stallman de création de la Free Software Foundation¹. Ces utilisateurs-experts sont à l'origine de nombreuses initiatives de projets de développement de logiciels libres (parmi lesquels Linux, Apache ou Samba). Dans le cas de ces logiciels phares, cette organisation a obtenu des résultats remarquables en terme de qualité et de rapidité d'amélioration, notamment parce que la libre disponibilité des sources permettait et permet toujours à ces utilisateurs-experts de tester les logiciels, d'étudier leur code et de corriger eux-mêmes les erreurs qu'ils y trouvent.

Les autres programmes libres largement distribués sont des outils de développement de programme (compilateurs, tels que GCC, un des premiers outils développé par la FSF, environnement de développement, comme Emacs ou Eclipse, dont IBM est à l'origine). Là encore, on s'adresse d'abord aux professionnels de l'informatique qui sont capables et intéressés par le développement ou l'adaptation de leurs outils de travail.

2.1.2 Le respect des normes

L'organisation coopérative du travail, le fait que les logiciels soient souvent une collection de petits projets évoluant simultanément, rendent également nécessaire que l'interface de communication entre ces projets/programmes soit publique et « normalisée » (c'est-à-dire respecte des formats publics dont l'évolution est définie collectivement).

Le code ouvert facilite la vérification de cette compatibilité et, si besoin est, la modification des programmes. Il est également remarquable de constater que, dans le but de ne pas renouveler les divergences de versions, ainsi qu'il en avait été pour Unix, les entreprises informatiques ont

1. Stallman a « inventé » le concept de Logiciel Libre, avec la création de la licence GNU / GPL et de la Fondation pour le Logiciel Libre (Free Software Foundation, FSF), l'organisme qui les produit ; voir <http://www.fsf.org>.

mis en place une organisation chargée de veiller à la compatibilité des différentes versions des distributions de Linux, la Linux Foundation, qui emploie entre autres le créateur de Linux, Linus Torvalds. Celle-ci publie, notamment, des recommandations techniques sur la façon de programmer les applications afin qu'elles puissent travailler avec ce système dans le même esprit que le standard POSIX, c'est-à-dire dans une optique de systèmes ouverts¹. Le fait que les entreprises utilisent des programmes libres peut être vu comme permettant de coordonner collectivement la création de composants (de « briques » logiciels) qui soient à la fois fiables et, surtout, « normalisés ». Jusqu'à présent, la création collective d'une telle base de composants « normalisés » a fait défaut à l'industrie informatique, même si des standards propriétaires, comme Windows, ont émergé².

Nous proposons maintenant de détailler comment les entreprises s'appuient sur ces organisations de production collective pour construire leur stratégie industrielle.

2.2 Des modèles basés sur les spécificités du Libre

Selon les secteurs industriels, comme nous l'avons montré³, le degré d'implication dans le logiciel libre est très varié, il s'appuie toutefois sur des motivations industrielles profondes et de long terme.

1. POSIX avait pour objet de proposer un ensemble d'interfaces standardisées permettant l'interopérabilité des différentes versions d'Unix entre elles et avec des logiciels d'application.

2. Gérard DRÉAN, *L'industrie informatique, structure, économie, perspectives*, Paris : Masson, 1996.

3. Nicolas JULLIEN et Jean-Benoît ZIMMERMANN, « Firms' contribution to open-source software and the dominant user's skill », dans : *European Management Revue* 6 (2009), p. 130–139.

Degré d'implication	Force motrice	Actif	Faible (assure la compatibilité)	Très faible à nulle
Logiciel	Logiciels techniques (Producteurs de composants logiciels et d'outils de développement) et architecteurs (Assembleurs de système d'information)	Éditeurs de systèmes d'exploitation (OS)		
Matériel		Serveurs	Portables de Haute Qualité (PC, portables de haut de gamme, PHQ), Outils de Communication Personnels (PDA, mobiles, OCP), (potentiel)	Portables à Bas Coût (PC portables d'entrée de gamme ¹ , PBC)

Degré d'implication des acteurs dans le logiciel libre, suivant les secteurs informatiques (Jullien & Zimmermann, 2009).

1. Nous restreignons au marché des portables qui est celui qui évolue le plus, et probablement, celui qui va s'imposer par rapport aux ordinateurs fixes. Mais le propos reste globalement vrai entre ordinateurs-tours de haut de gamme et les machines d'entrée de gamme.

Ces différences s'expliquent par les caractéristiques de ces sous-marchés, particulièrement le régime de concurrence et les compétences des clients-utilisateurs des produits.

Tous les segments sont caractérisés par des régimes de concurrence fortement imparfaits, dus à divers « rendements croissants d'adoption »¹ : économies d'échelle en production, forts coûts fixes, notamment dus à la R&D, interrelations technologiques, effets d'apprentissage). Un seul marché peut être considéré comme quasiment monopolistique (celui des systèmes d'exploitation pour les PC), les autres étant plutôt des oligopoles, avec, dans tous les cas, de fortes barrières à l'entrée.

À l'intérieur de ces différents oligopoles, les stratégies de différenciation jouent généralement un rôle important. Le marché des ordinateurs à bas coûts (PBC), caractérisé par une concurrence en prix, est un marché relativement neuf, une différenciation verticale (par les prix) du marché traditionnel des PC. Dans les autres cas, les stratégies industrielles reposent plutôt soit sur de la différenciation horizontale (intégration de nouvelles caractéristiques ou outils, comme dans les téléphones mobiles), soit sur de la segmentation de marché grâce aux formats propriétaires et des stratégies d'intégration matériel-logiciel-contenu, comme dans les consoles de jeu.

Dans de tels marchés, un nouvel entrant doit proposer une différenciation significative, soit verticale, comme un meilleur ratio performance/prix (des portables performants moins chers ou des serveurs/des portables de haute qualité aux capacités de calcul supérieures pour le même prix), soit horizontale (des caractéristiques nouvelles, valorisées par au moins une partie des utilisateurs, comme l'a fait Samsung sur le marché des téléphones mobiles avec le téléphone à clapet).

Sur ces marchés technologiques, et particulièrement sur le logiciel, les utilisateurs ont un double rôle, fonction de leur potentiel économique (pouvoir d'achat, ou de marché), mais aussi technique (capacité à comprendre et à intervenir sur la solution développée). D'un côté du spectre, les clients des grands intégrateurs, comme Cap Gemini ou IBM, que nous appellerons par la suite les « architectes » de services, sont souvent de

1. W. Brian ARTHUR, « Competing technologies, increasing returns and lock-in by historical events : The dynamics of allocations under increasing returns to scale », dans : *Economic Journal* 99 (1989), p. 116-131, URL : <http://www.santafe.edu/arthur>.

	Monopole	Oligopole basé sur les prix	Oligopole horizontal
Naïf	OS pour portable	Portables bas coût	lecteurs
Kogut-Metiu (KM)			Outils de communication personnels
	OS pour serveurs PC		Portables haute qualité
Von Hippel (VH)			Serveurs
			Logiciels techniques et architecteurs

Les sous-marchés des TIC en fonction des capacités techniques des utilisateurs et du régime de concurrence. (Jullien & Zimmermann 2009)

grandes entreprises, avec de fortes compétences techniques. À l'autre extrémité, les vendeurs de portables d'entrée de gamme s'adressent à un marché de masse où chaque individu a un très faible pouvoir de marché et peu de compétences en moyenne. En croisant les compétences techniques (des utilisateurs « naïfs », sans compétence, aux utilisateurs « von Hippel » (VH), en référence à von Hippel¹, c'est-à-dire des utilisateurs-innovateurs, capables de co-développer les technologies, en passant par les utilisateurs « Kogut-Metiu » (KM), exigeants techniquement, mais ne développant pas² et le type de marché on obtient une segmentation plus fine des différents sous-marchés.

Les compétences de ces utilisateurs, et leur pouvoir de marché, expliquent l'intégration de logiciels libres et l'implication des producteurs dans le développement des logiciels libres, motivés par une différenciation significative (le plus souvent verticale). Celle-ci peut permettre, pour les entrants, de surmonter les barrières à l'entrée créées par le fonctionnement oligopolistique ou, au contraire, pour les firmes installées, de perpétuer leur position sur le marché.

Ainsi, les éditeurs de distribution Linux ont été parmi les premiers acteurs commerciaux à entrer sur le marché des systèmes d'exploitation

1. Eric von HIPPEL, *The Sources of Innovation*, Oxford : Oxford University Press, 1994.
 2. Bruce KOGUT et Anca METIU, « open Source Software Development and Distributed Innovation », dans : *Oxford Review of Economic Policy* 17.2 (2001), p. 248–264, URL : <http://oxrep.oxfordjournals.org/content/17/2/248>.

pour PC en utilisant les logiciels libres (1995 pour Red Hat). Cela semblait une évidence. Il s'agit d'un marché de masse, avec une concurrence basée significativement sur les prix. En fait, peu d'utilisateurs sont assez compétents pour installer un OS¹ alternatif (les utilisateurs naïfs y sont majoritaires), et les machines sont achetées avec un OS déjà installé et payé. Le marché ne s'est pas développé pour l'instant. Les producteurs de portables pourraient changer la donne, pour diminuer la position dominante de Microsoft. Sur le marché des portables à bas coût, installer un OS libre peut être considéré comme une manière de réduire les prix, donc d'augmenter la part de marché de l'entreprise et donc d'élargir la base de marché. Mais l'initiative viendrait du constructeur et non de l'utilisateur. Comme, dans ce cas, la raison pour utiliser du logiciel libre est de proposer le prix le plus bas possible, les entreprises n'investiront pas dans le développement plus que l'effort nécessaire pour adapter le logiciel à leur(s) produit(s) : cela augmenterait leur coût. Dans ce cas, le logiciel libre peut être considéré comme un bien gratuit, un *freeware*. Bien sûr, on pourrait avancer que ces stratégies ont un caractère opportuniste² et le risque est grand de démotiver les personnes les plus engagées dans les communautés, qui constatent les bénéfices que d'autres réalisent en puisant dans leur travail volontaire³. Toutefois, en adoptant des produits FLOSS, comme le système d'exploitation Linux, la suite bureautique Libre Office, ou le navigateur Web Firefox, ces entreprises participent à l'expansion du réseau des utilisateurs. Dans un régime de concurrence dans laquelle la bataille pour les externalités de réseau et les normes joue un rôle crucial, cela peut donner une forte impulsion aux logiciels libres, permettant à ces entreprises de gagner au moins la neutralité de la communauté si ce n'est son approbation.

Sur le marché alors émergent du système d'exploitation pour serveurs PC, les choses ont été différentes. La plupart des utilisateurs (de type VH ou KM) savaient installer et configurer un système d'exploitation. Un sys-

1. *Operating System* (système d'exploitation).

2. On appelle stratégie opportuniste ou comportement opportuniste (*free-riding* en anglais) le fait de bénéficier des contributions, des efforts des autres sans apporter sa propre contribution.

3. Dominique FORAY, Sylvie THORON et Jean-Benoît ZIMMERMANN, « Open software : knowledge openness and cooperation in cyberspace », dans : *Internet and digital economics : Principles, methods and applications*, sous la dir. d'Eric BROUSSEAU et Nicolas CURIEN, New York : Cambridge University Press, 2007, p. 368–390.

tème libre leur donnait accès à un système d'exploitation du type Unix, mais moins cher, plus ouvert et plus adaptable que ce qu'ils pouvaient trouver sur le marché Unix traditionnel où ils s'approvisionnaient habituellement. Cela a permis aux éditeurs de systèmes d'exploitation libres d'entrer sur le marché (mais ce n'est pas le même marché). Entre les deux, on trouve le marché des portables haut de gamme (PHQ), où les utilisateurs sont plus compétents que sur le marché PBC, mais avec un système d'exploitation pré-installé sur la machine. Le constructeur reste l'acteur qui pourra décider du changement, en tenant compte, le cas échéant, des demandes des utilisateurs (cf. l'exemple de DELL qui explore l'installation et la vente de machines avec Linux installé de façon « native », sous la pression/demande de ses utilisateurs). Mais, là encore, ce n'est pas l'éditeur de la distribution qui fera évoluer ce marché, c'est le producteur ou le distributeur du matériel.

Quand on regarde la colonne « oligopole horizontal », qui est celle des producteurs de matériels, il est clair que plus les utilisateurs sont qualifiés, plus le concept de logiciel libre et les offres industrielles correspondantes ont des chances de diffuser.

À un extrême, les formats propriétaires ont pu maintenir une intégration forte entre matériel-logiciel-contenu sur le segment des consoles de jeu, parce que les jeux développés pour Linux n'ont pas toujours la qualité de ceux développés pour les standards propriétaires (PS2/3, Xbox...). C'est aussi le cas, à un niveau un peu plus faible sur le segment des « players » musicaux : sur ce segment, le standard MP3 (ou Ogg) fait qu'une nouvelle entrée est toujours possible.

À l'autre extrême, on (re-)trouve le marché des serveurs, où les producteurs proposaient des solutions propriétaires (matériel et Unix propriétaires liés). Sur ce marché, les fournisseurs travaillent avec des utilisateurs VH, qui peuvent être des contributeurs de premier ordre dans le contexte d'une ouverture libre. Ainsi, le développement des serveurs PC a permis à certains de ces utilisateurs de contourner le problème de l'intégration matériel-logiciel (*bundling*) ; utiliser Linux permettait de construire une solution moins onéreuse (avantage vertical), en réutilisant le portefeuille des programmes Unix (existence d'un contenu). Sur la base de la révéla-

tion de cette demande, certaines entreprises, comme Cobalt¹, ont cherché à élargir ce marché des serveurs PC sous Unix, vers les utilisateurs KM, sensibles au prix, mais aussi à la qualité d'un serveur PC, utilisant Linux. Rapidement, les acteurs majeurs d'Unix, pour les raisons expliquées plus haut, ont aussi développé de telles offres, limitant par là même les possibilités de différenciation verticale, donc les entrées.

Les outils personnels de communication (aujourd'hui les smartphones) présentent un cas intermédiaire avec des utilisateurs moins « techniques » (KM+N), et un degré d'implication des acteurs commerciaux faible, seulement préoccupés par des questions de compatibilité. Depuis le début des années 2000, on assiste à une ouverture des architectures (et des fonctionnalités) sur ce marché, en parallèle au système intégré proposé par Apple avec son iPhone. Il semble que le cœur du métier (et de la différenciation) de ces constructeurs ne soit pas le système d'exploitation, mais la maîtrise du design et de la production du matériel, intégrant toujours plus de fonctions, ainsi que l'ergonomie logicielle et matérielle des services proposés. En l'absence d'un standard de fait sur le système d'exploitation, Linux, ou plutôt son adaptation par Google, Android, apparaît comme un choix rationnel pour les constructeurs : il est gratuit, sous licence libre, et soutenu par une communauté d'utilisateurs-développeurs (externalisation d'une partie de la R&D et du développement), même si Google reste prépondérant dans le développement. Enfin, l'ouverture et le respect des standards fait qu'il est plutôt facile à adapter. Cela permet ainsi aux constructeurs de réduire leurs coûts et de se concentrer sur leur cœur de métier. Il est amusant de constater que les acteurs des smartphones semblent rejouer le régime de concurrence des ordinateurs personnels des années 1980, cette fois-ci avec un système libre.

Pour finir, sur les marchés logiciels où les utilisateurs sont plutôt de type VH (outils techniques et architecteurs), beaucoup d'entreprises se sont déjà tournées vers le logiciel libre. Cela représente un moyen efficace pour elles de se différencier verticalement, en termes de performances (qualités techniques du produit, meilleur système de gestion des retours utilisateurs...). Ces stratégies ont été initiées par des entreprises qui n'occupaient pas une position dominante sur le marché, nouveaux entrants

1. Cobalt a été racheté par SUN, mais l'intégration de cette offre n'a pas suffi pour contrer l'invasion des PC sous GNU/Linux, ce qui a conduit au rachat de SUN par Oracle.

(comme MySQL dans les bases de données) ou en difficulté (comme IBM avec l'outil de développement Eclipse). La stratégie libre leur a permis de réduire les coûts de développement en intégrant les contributions des utilisateurs VH, mais aussi de diminuer leurs coûts marketing en tirant parti de la dynamique de diffusion virale dans cette population d'utilisateurs qualifiés.

Type d'utilisateur	Régime de concurrence	Acteurs/ « produits »	Degré d'implication	Motivations pour se tourner vers le logiciel libre	But
VH	Oligopole avec différenciation horizontale.	Logiciels techniques et architecteurs	Force principale	Qualité intrinsèque et assurée dans le temps des logiciels + signal	Entrée (MySQL) parts de marché (différenciation verticale)
VH	Oligopole « clivé »	Constructeurs de serveurs	Actif	Qualité & ouverture dans le monde Unix	Réponse au développement des serveurs PC (sous Windows)
KM	Monopole fort	Éditeurs de systèmes d'exploitation	Actif	Capacité à proposer un système d'exploitation Unix + Standards ouverts	Contournement des barrières à l'entrée
KM	Oligopole avec différenciation horizontale.	Constructeurs de portable de haut de gamme	Faible (au niveau des capacités d'absorption)	Qualité + outils, applications ouvertes	Différenciation verticale (prix/performances)
KM+N	Oligopole avec différenciation horizontale.	Outils de Communication Personnels	Faible (compatibilité)	Actifs complémentaires	Permet de se concentrer sur les compétences « cœurs »
N ¹	Oligopole « clivé »	Lecteurs (de musique)	Faible (compatibilité)	Délier matériel, logiciel et contenu	Contournement des barrières à l'entrée
N	Oligopole basé sur les prix	Constructeurs de portable de bas de gamme	Faible à nul	Prix	Concurrence en prix

Le rôle décisif des utilisateurs. Source : à partir de Julien-Zimmermann (2009)

1. Mais VH nécessaire pour avoir des contenus ouverts.

2.3 Analyse

Le premier modèle est celui de la coopération autour d'un logiciel « plate-forme », ce que Romer¹ a appelé un « bien public industriel », comme Linux, ou Apache. Ces logiciels jouent un rôle clef dans la production des produits informatiques auxquels ils fournissent un socle. À cause des interrelations technologiques, chaque constructeur de matériel ou de logiciel doit s'assurer que ces logiciels sont compatibles avec les produits qu'il développe. La coopération prend alors le plus souvent la forme d'un consortium dont l'objectif est de coproduire les briques de base au coût le plus faible, et de veiller à ce qu'elles soient adaptées par chaque participant à leurs besoins, tout en assurant une cohérence d'ensemble (production d'une norme).

Cette logique correspond à l'origine à une situation de collaboration entre pairs, entre professionnels de l'informatique qui ont initié une telle pratique. Mais elle réunit de plus en plus d'entreprises, avec les investissements d'IBM, de Red Hat, ou de SUN/Oracle, dans Linux ou Apache. Notons aussi que, pour la plupart de ces plates-formes, des « fondations » dédiées participent étroitement au contrôle du projet².

Le second modèle est celui de la « capacité technique entretenue », pour reprendre les analyses de Jean Gadray³. Autour d'un outil technique innovant, évolutif, le producteur (marchand) organise la collaboration avec ses utilisateurs-innovateurs (entreprise clientes, développeurs), bref des utilisateurs sources d'innovation, au sens de Von Hippel⁴. Dans une relation « symbiotique » pour reprendre les termes de Dahlander et Magnusson⁵, le producteur agrège les contributions, assure la stabilité de l'outil, aide les développeurs à l'utiliser. Les utilisateurs-développeurs (éventuellement des entreprises utilisant l'outil), en fournissant leurs innovations, sont ainsi assurés que leurs besoins seront pris en compte plus rapidement,

1. Paul M. ROMER, « The economics of new ideas and new goods », dans : *Annual Conference on Development Economics*, Washington D. C. : Banque Mondiale, 1993.

2. Ce sont l'Apache Foundation pour Apache (<http://www.apache.org>), l'Internet Systems Consortium pour des logiciels comme BIND (serveur de DNS, <http://www.isc.org>), La Linux Foundation pour Linux (<http://www.linux-foundation.org>)...

3. Jean GADRAY, *Socio-économie des services*, Paris : La Découverte, 2003.

4. HIPPEL, *op. cit.*

5. Linus DAHLANDER et Mats G. MAGNUSSON, « How do Firms Make Use of Open Source Communities ? », dans : *Long Range Planning* 41 (2008), p. 629-649.

et intégrés-maintenus dans le produit (ce qui est un point fondamental de la diminution de leur coût selon Von Hippel). Le producteur vend ce que nous avons appelé les « trois A » (assurance qualité, adaptation (rapide) aux besoins des utilisateurs, et assistance à l'utilisation¹). MySQL AB, producteur de logiciel libre de bases de données est un archétype de ce type d'acteur. Ce modèle n'est pas en soi très différent du modèle traditionnel des producteurs d'outils informatiques, comme Oracle. Mais l'ouverture permet sans doute un meilleur retour de la part des utilisateurs, et donc une plus forte externalisation des coûts de développement, tout en assurant une évolution plus rapide du logiciel. Comme l'actif spécifique de ces entreprises repose sur la gestion de l'édition de logiciels, celles-ci doivent investir dans le logiciel qu'elles modifient, et l'implication des développeurs salariés de l'entreprise est souvent concentrée sur son produit (libre).

Entre ces deux stratégies, on trouve les entreprises de services. La seule incertitude de l'intégration des logiciels libres dans leur modèle d'affaires découle de la disponibilité et la qualité des composants, qualité qu'ils doivent être capable d'évaluer, et qui passe par une participation aux communautés de développement, comme l'ont montré Dahlander et Magnusson². Cette nécessité d'une évaluation et de contrôle va croissante avec l'importance du composant dans les solutions proposées par l'entreprise ou demandées par ses clients. On peut même formuler l'hypothèse que plus les clients sont qualifiés, plus l'entreprise doit maîtriser la technologie, en raison du niveau croissant de complexité des rétroactions et de la demande. C'est ce que Dahlander et Magnusson³ ont appelé une « approche commensalistique ». Cela correspond à l'affirmation plus générale de Cohen et Levinthal⁴ sur la nécessité pour une entreprise de faire des efforts internes de R & D une condition préalable à l'absorption de technologie extérieure.

Nous considérons que cela reflète un changement dans les technologies utilisées, donc dans les actifs complémentaires que ces entreprises

1. JULLIEN et ZIMMERMANN, *op. cit.*

2. DAHLANDER et MAGNUSSON, *op. cit.*

3. Linus DAHLANDER et Mats G. MAGNUSSON, « Relationships Between Open Source Software Companies and Communities : Observations from Nordic Firms », dans : *Research Policy* 34 (2005), p. 481–493.

4. Wesley M. COHEN et Daniel A. LEVINTHAL, « Innovation and Learning : The Two Faces of R&D », dans : *Economic Journal* 99 (1989), p. 569–596.

ont besoin de gérer, mais pas vraiment dans leurs compétences de base. Les entreprises « architecteurs » traditionnelles ne sont pas forcément impliquées dans les développements libres car elles n'utilisent pas ces technologies. Mais elles ont d'autres processus, comparables, pour surveiller l'évolution de l'actif complémentaire, les technologies qu'elles utilisent. Ainsi, elles peuvent participer à des sessions de formation des éditeurs, ou conclure des « alliances mondiales » avec leurs principaux partenaires, comme le fait Cap Gemini ¹.

Dans le deuxième modèle, il semble que la licence joue un rôle important : l'adoption de la licence GNU GPL est un engagement fort du producteur à ne pas s'approprier les contributions des utilisateurs, et à en faire profiter tous les utilisateurs. Une autre licence pourrait générer un certain doute et limiter la confiance et l'implication des développeurs. Ainsi le projet de navigateur libre, basé sur Netscape n'a vraiment fonctionné que lorsque l'entreprise a publié son code sous une licence compatible avec la GNU GPL et la licence du logiciel Qt (QPL), édité par Trolltech, a évolué vers la GNU GPL sous la pression des développeurs-utilisateurs. Dans ce modèle, paradoxalement, la GNU GPL apparaît aussi un moyen de contrôler la concurrence : si un concurrent veut s'approprier le logiciel pour en faire une version différente, il lui faudra publier ses améliorations, qui pourront être aussitôt intégrées dans le logiciel original.

Dans le premier modèle, la licence semble jouer un rôle moins important. Certains logiciels, comme Apache, ou Bind ne sont pas protégés par des licences obligeant le reversement des contributions, comme la GNU GPL. Cela peut sembler paradoxal, car dans le cas de la collaboration entre entreprises, la GNU GPL pourrait renforcer cette collaboration, rendant plus difficile la défection. Le choix de ces licences s'explique par le fait que ces logiciels ont souvent été produits par les universités américaines, et donc traditionnellement mis à disposition de tous (licences BSD). D'autre part, le fait que ces logiciels soient co-développés rend plus difficile le changement de licence (qui a le pouvoir de le décider ?) En réalité, cette question du choix de licence n'a que peu d'impact sur la coopération car la dynamique d'innovation suffit à rendre la défection trop coûteuse : faire défection signifie assurer soi-même la mise au point et la maintenance de ses propres améliorations, vérifier à chaque fois qu'il y a

1. <http://www.capgemini.com>.

une nouvelle version qu'elles restent compatibles avec celle-ci, et risquer de devoir faire face à des choix incompatibles avec les produits que l'on vend.

Conclusion

Il faut souligner que l'émergence de ce modèle du logiciel libre signifie l'intégration de l'ensemble de la chaîne de production, de la chaîne de valeur, dans le développement des logiciels : pas seulement les producteurs-vendeurs, mais aussi les utilisateurs-développeurs, les entreprises qui intègrent le composant dans leur produit, et la recherche. Ceci est rendu possible grâce à une structuration forte de l'organisation de production, basée sur des choix techniques (production modulaire), et soutenu par une utilisation originale du droit d'auteur (à travers des licences spécifiques). Ensuite, selon le segment ou le type de logiciel concerné, la place du logiciel dans la chaîne de production, la place des acteurs privés, et celle des utilisateurs peut fortement varier, et donc l'importance du recours au libre, mais en fonction de préoccupations industrielles et marketing classiques. Le renouveau de la coopération dans le développement logiciel, comme sa quasi-disparition dans les années 1970, s'explique par les mêmes mécanismes, évolution de la demande, évolution des technologies, qui changent le modèle de production industriel. Ce mode d'organisation de production libre qui s'affirme aujourd'hui sur des bases renouvelées apparaît bien comme celui du « monde Internet », tel qu'il a commencé à s'esquisser autour du concept du Web 2.0.

Références

- ARTHUR, W. Brian, « Competing technologies, increasing returns and lock-in by historical events : The dynamics of allocations under increasing returns to scale », dans : *Economic Journal* 99 (1989), p. 116–131, URL : <http://www.santafe.edu/arthur>.
- COHEN, Wesley M. et Daniel A. LEVINTHAL, « Innovation and Learning : The Two Faces of R&D », dans : *Economic Journal* 99 (1989), p. 569–596.

- DAHLANDER, Linus et Mats G. MAGNUSSON, « How do Firms Make Use of Open Source Communities ? », dans : *Long Range Planning* 41 (2008), p. 629–649.
- « Relationships Between Open Source Software Companies and Communities : Observations from Nordic Firms », dans : *Research Policy* 34 (2005), p. 481–493.
- DANG NGUYEN, Godefroy et Christian GENTHON, « Les perspectives du secteur des TIC en Europe », dans : *Cahiers de Recherche Marsouin* 4 (2006), URL : <http://www.marsouin.org>.
- DRÉAN, Gérard, *L'industrie informatique, structure, économie, perspectives*, Paris : Masson, 1996.
- FITZGERALD, Brian, « The transformation of open source software », dans : *MIS Quarterly* 30.3 (2006), p. 587–598, URL : <http://ulir.ul.ie/handle/10344/92>.
- FORAY, Dominique, Sylvie THORON et Jean-Benoît ZIMMERMAN, « Open software : knowledge openness and cooperation in cyberspace », dans : *Internet and digital economics : Principles, methods and applications*, sous la dir. d'Eric BROUSSEAU et Nicolas CURIEN, New York : Cambridge University Press, 2007, p. 368–390.
- GADRAY, Jean, *Socio-économie des services*, Paris : La Découverte, 2003.
- GENTHON, Christian, *Croissance et crise de l'industrie informatique mondiale*, Paris : Syros, 1995.
- GENTHON, Christian et Denis PHAN, « Les logiciels libres : un nouveau modèle ? », dans : *Terminal* 80 (1999), p. 167–188.
- GRAY, Pamela A., *Open systems : a business strategy for the 1990's*, London : McGraw-Hill, 1991.
- HIPPEL, Eric von, *The Sources of Innovation*, Oxford : Oxford University Press, 1994.
- HORN, François, *L'économie des logiciels*, Paris : La Découverte, 2004.
- JULLIEN, Nicolas et Jean-Benoît ZIMMERMANN, « Firms' contribution to open-source software and the dominant user's skill », dans : *European Management Review* 6 (2009), p. 130–139.
- KOGUT, Bruce et Anca METIU, « open Source Software Development and Distributed Innovation », dans : *Oxford Review of Economic Policy*

- 17.2 (2001), p. 248–264, URL : <http://oxrep.oxfordjournals.org/content/17/2/248>.
- LAKHANI, Karim R. et Robert G. WOLF, « Why Hackers Do What They Do : Understanding Motivation and Effort in Free/Open Source Software Projects », dans : *Perspectives on Free and Open Source Software*, sous la dir. de Joseph FELLER et al., Cambridge, Mass. : The MIT Press, 2007, p. 3–22.
- LANGLOIS, R. N. et D. C. MOWERY, « The Federal Government Role in the Development of the U.S. Software Industry : An Assessment », dans : *The International Computer Software Industry : A Comparative Study of Industry Evolution and Structure*, sous la dir. de D. C. MOWERY, Oxford : Oxford University Press, 1996.
- MCKUSICK, Marshall K., « Twenty Years of Berkeley Unix. From AT&T-Owned to Freely Redistributable », dans : *Open Sources : Voices from the Open Source Revolution*, sous la dir. de Chris DIBONA, Sam OCKMAN et Marc STONE, Sebastopol, CA : O'Reilly, 1999.
- POUZIN, Louis, « The Cyclades computer network », dans : *ICCC 2* (1982).
- ROMER, Paul M., « The economics of new ideas and new goods », dans : *Annual Conference on Development Economics*, Washington D. C. : Banque Mondiale, 1993.
- ROSENBERG, Nathan, *Inside the Black Box : Technology and Economics*, Cambridge, MA : Cambridge University Press, 1983.
- ZIMMERMANN, Jean-Benoît, « L'industrie du logiciel, de la protection à la normalisation », dans : *Changement institutionnel et changement technologique : évaluation, droits de propriété intellectuelle, système national d'innovation*, sous la dir. de Baslé DUFOUR, CNRS Editions, Paris, 1995, p. 181–207.

Les RMLL, haut lieu mobile du libre francophone

Pierre-Amiel GIRAUD

Les Rencontres Mondiales du Logiciel Libre (RMLL) sont un ensemble de conférences, d'ateliers de démonstration et de formation, dont le but premier est de faire se rencontrer les acteurs du Libre mais aussi de sensibiliser le grand public et les collectivités territoriales. Elles ont lieu chaque année, depuis 2000, au début du mois de juillet. Après avoir rassemblé quelque 500 personnes pour leur première édition, leur fréquentation se stabilise entre 4000 et 5000 participants à partir de 2008. Leur réputation parmi les militants du Libre dépasse l'espace linguistique francophone, bien que ce dernier soit de loin le plus représenté. Leur succès, sans cesse croissant, vient interroger certaines représentations courantes touchant le fonctionnement et l'organisation du Libre.

Bien souvent, en effet, les modèles collaboratifs qui peuvent être mis en œuvre induisent la mobilisation de communautés dont les membres sont dispersés : les « communautés distantes »¹. Le Libre semble ainsi appuyer la thèse d'économistes pour qui Internet annoncerait la fin de

1. Nicolas JULLIEN, Didier DEMAZIÈRE et François HORN, « How free software developers work. The mobilization of *distant communities* », dans : *Cahier de recherches* 7 (2006), URL : http://www.marsouin.org/IMG/pdf/Demaziere-Horn-Jullien_7-2006.pdf.

la géographie, les questions de localisation perdant de leur importance¹. Les géographes ont tôt fait de décrier ces affirmations². D'ailleurs, l'existence et le succès des RMLL montrent, sans nier l'importance des transformations de la géographie induites par le numérique, que le cyberspace ne vient pas se substituer à l'espace géographique antécédent, mais qu'il vient l'enrichir de nouvelles potentialités relationnelles³. Les RMLL ne sont donc pas une anomalie dans une communauté désincarnée et insaisissable. Elles sont au contraire un *lieu* important d'une mouvance inscrite dans la société et impliquée dans les problèmes publics⁴ de son temps.

Le lieu est l'un des concepts fondamentaux en géographie. Il en existe de nombreuses définitions concurrentes, parfois incompatibles. L'une des plus pertinentes, c'est-à-dire des plus efficaces pour interpréter l'espace géographique de la mouvance du Libre, est celle selon laquelle un lieu est un « espace dans lequel la distance n'est pas pertinente »⁵. Il n'y a donc pas de lieu en soi, mais seulement relativement à certains objectifs, à certaines représentations. Cela signifie qu'au moins deux réalités doivent être en contact pour qu'il y ait lieu ; c'est-à-dire justement qu'elles se rencontrent. Cette définition interdit de considérer le lieu comme un donné (point d'appui jamais discuté d'un raisonnement), et oblige à le comprendre comme une construction spatiale du social.

Dans quelle mesure peut-on éclairer réciproquement : d'une part, la ou les formes spatiales des RMLL ; d'autre part, les stratégies et les représentations des acteurs ? Que nous dit cet objet géographique de l'espace dans lequel vivent les libristes qui y participent ?

Nous avons obtenu les données nécessaires à la rédaction de ce texte en croisant plusieurs pratiques assez courantes en sciences sociales. D'abord, conformément à la méthode de l'observation participante, nous avons pris

1. Richard O'BRIEN, *Global financial integration : the end of geography*, London : Royal Institute of International Affairs ; Chatham House papers, 1992.

2. Frédéric LASSERE, « Internet : la fin de la géographie ? », dans : *Cybergeo : European Journal of Geography* (2000), URL : <http://cybergeo.revues.org/4467>.

3. Boris BEAUDE, *Internet : changer l'espace, changer la société : les logiques contemporaines de synchronisation*, Limoges : Fyp éditions, 2012.

4. Elizabeth SHEPPARD, « Problème public », dans : *Dictionnaire des politiques publiques*, Paris : Presses de Sciences Po., 2010, p. 530-538.

5. Augustin BERQUE, « Lieu », dans : *Dictionnaire de la géographie et de l'espace des sociétés*, sous la dir. de Jacques LÉVY et Michel LUSSAULT, Paris : Belin, 2003, p. 555-556.

part activement à plusieurs éditions des RMLL afin de les vivre de l'intérieur. Ensuite, nous avons interviewé plusieurs participants et organisateurs pour savoir les mots et les sens qu'ils mettent sur leurs pratiques aux RMLL. Enfin, nous avons mis à profit la documentation disponible en ligne, par exemple sur les sites des différentes éditions de la manifestation.

Il ressort de ces données qu'après une phase d'ancrage en Aquitaine et plus précisément à Bordeaux, les RMLL sont devenues un lieu mobile. De plus, leur place symbolique dans la représentation des libristes francophones en fait aussi un haut lieu, dont l'un des rôles est de servir à la mouvance de point d'entrée sur les territoires qui l'accueillent.

1. Les RMLL, de l'ancrage aquitain au lieu mobile

L'initiative des RMLL revient à l'ABUL (alors Association Bordelaise des Utilisateurs de Linux¹), l'un des GULL (Groupe d'Utilisateurs de Logiciels Libres) français les plus anciens, fondé en 1999. Très vite, il cherche à sensibiliser les élus locaux afin de pouvoir, par leur intermédiaire, pénétrer plus efficacement le territoire aquitain. Ainsi, dès le mois de décembre, le projet des RMLL est lancé, comme en témoigne un courriel de Pierre Jarillon sur la liste de diffusion de l'ABUL² qui peut être considéré comme l'acte de naissance de l'événement. Après trois éditions à Bordeaux (2000, 2001 et 2002) sous l'égide de ce GULL, les RMLL se déplacent dans diverses villes françaises, pour finalement sortir de l'Hexagone en 2012. Pourtant, où que soit l'hôte, le même lieu se déploie toujours : les RMLL sont un lieu mobile.

Les RMLL ne sont pas la seule manifestation de la mouvance du Libre. Cependant, elles comptent parmi les plus anciennes et sont les seules à se réclamer à la fois du niveau mondial, à être organisées par des associations, et à associer le grand public. Si la première conférence *hacker* à proprement parler a eu lieu en 1984 à San Francisco, elle était purement américaine et ne faisait pas référence au logiciel libre³. Dans la

1. Elle devient Association Bordelaise des Utilisateurs de Logiciels Libres lors de l'assemblée générale du 15 janvier 2000.

2. Une copie de ce texte est disponible sur le site des RMLL 2010 : <http://2010.rml.info/Il-y-a-dix-ans-naissaient-les-Rencontres-Mondiales-du-Logiciel-Libre.html>.

3. F. TURNER, « How Digital Technology Found Utopian Ideology : Lessons From the First Hackers' Conference », dans : *Critical Cyberculture Studies : Current Terrain, Future*

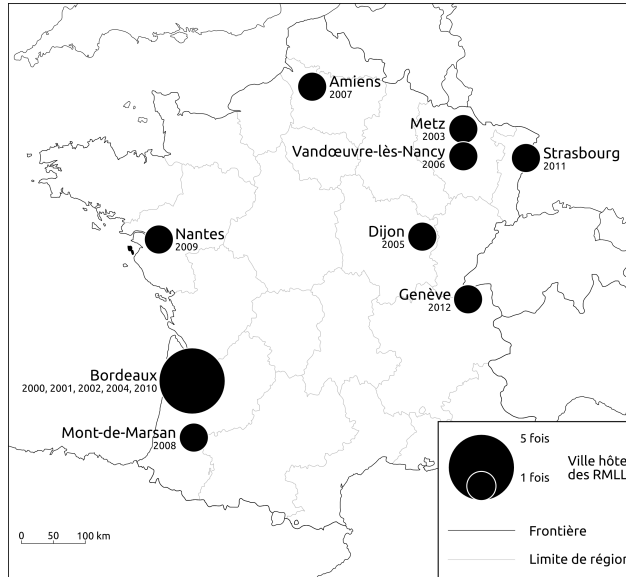
deuxième moitié des années 1990, des conférences autour de Linux sont organisées, comme le *Linux World Show*. Il s'agit cependant de conférences d'affaires, dans le cadre de la bulle Internet. Le besoin de faire des conférences globales capables de mobiliser l'ensemble de la mouvance ne s'est pas fait sentir seulement de ce côté de l'Atlantique : en parallèle des RMLL, sans concertation et sans connaissance réciproque des projets, des Brésiliens fondent le *Fórum Internacional do Software Livre*. Depuis, les manifestations en tous genres autour du Libre se sont multipliées. Par exemple, tout projet logiciel un tant soit peu notable possède au moins une conférence annuelle. En France, on peut noter l'existence de l'*Open World Forum*, qui se tient tous les ans au début de l'automne à Paris depuis 2008. Cet événement est cependant beaucoup plus orienté vers le monde des affaires et des entreprises.

Après avoir décrit et explicité les implications de cette réalité, nous montrerons que d'une certaine manière les RMLL restent très liées à leur territoire natal par l'intermédiaire de leurs fondateurs. Enfin, nous montrerons que cette mobilité masque certains ancrages implicites.

1.1 Plusieurs sites, un seul lieu : les RMLL sont un lieu mobile

Au total, neuf villes ont hébergé les RMLL. Notons que les RMLL bordelaises ont lieu surtout (voire exclusivement les premières années) sur le campus de Talence. On ne retrouve pas cette imprécision pour les autres éditions. Il y a parfois un découplage entre la ville qui donne son nom à la manifestation et la commune qui l'accueille effectivement. C'est le cas par exemple des RMLL de Bordeaux, qui techniquement ont surtout lieu sur la commune de Talence. Dans le but d'attirer des conférenciers lointains, il pouvait être préférable d'afficher un nom mondialement connu. À l'inverse, l'édition de 2006 porte le nom de la commune qui l'héberge. Il faut dire que Vandœuvre-lès-Nancy en est le principal financeur et qu'elle soigne une image de commune en pointe sur le numérique. En effet, elle est alors l'une des trois seules municipalités françaises à détenir les cinq @ du label Ville Internet. En outre, les RMLL sont à cette date une manifestation bien connue de la mouvance du Libre, au-delà même de

Directions, 2006 ; Gabriella COLEMAN, « The Hacker Conference : A Ritual Condensation and Celebration of a Lifeworld », dans : *Anthropological Quarterly* 83.1 (2010), p. 47–72.



Les RMLL : un seul lieu, des sites multiples

l'espace francophone : l'identité de la ville hôte est moins critique, et n'est pertinente qu'au regard des infrastructures qu'elle est susceptible d'offrir.

Alors même que les RMLL sont aujourd'hui un lieu mobile, l'espace géographique dessiné par les Rencontres se différencie, à la fois de manière explicite par les modalités de sélection mises en œuvre par le comité des RMLL, et de manière plus implicite à travers ce qui peut apparaître comme des critères tacites de sélection.

1.1.1 Comment un lieu peut-il être mobile ?

Puisqu'il n'y a de lieu qu'en fonction d'intentions et de représentations, on ne peut pas fixer de taille maximale au lieu, au-delà de laquelle il faudrait nécessairement parler de surface. Au contraire, le monde lui-même peut apparaître comme un lieu¹, par exemple quand il est envisagé sous l'angle du réchauffement climatique. L'inverse est également vrai : même

1. Denis RETAILLÉ, « La transformation des formes de la limite », dans : *Articulo. Journal of Urban Research* 6 (2011), URL : <http://articulo.revues.org/1723>.

au sein d'un tout petit espace la distance peut être pertinente. Qui n'a jamais été trop paresseux pour aller chercher un dictionnaire sur une étagère située à 1,50 m de son bureau ?

Deuxièmement, elle permet d'envisager des lieux sans localisation fixe. Puisque le lieu est défini par le contact entre deux ou plusieurs réalités, la localisation topographique de ce dernier ne change rien à l'identité du lieu qui se manifeste – qui, précisément, a lieu. C'est ainsi qu'il faut comprendre l'expression « aller aux RMLL » voire aux « ReuMeuLeuLeu », très employée par les participants : l'identité du lieu perdure malgré les changements de ville hôte et d'équipe organisatrice.

La ville d'accueil n'est pas pour autant dépourvue de statut géographique. La théorie de l'espace mobile¹, qui la disqualifie comme lieu relativement aux RMLL, apporte également la notion de site, désignant ainsi simplement ce qui « accueille » ces « circonstances plus ou moins durables » que sont les lieux².

La distinction d'avec les sites permet donc d'assurer la permanence du lieu malgré une localisation mouvante. Elle ne permet pas, cependant, de comprendre comment les RMLL maintiennent leur identité malgré une équipe organisatrice propre à chaque ville hôte. C'est plutôt du côté des modalités d'organisation de l'événement qu'il faut aller chercher des réponses.

1.1.2 La sélection des sites par le comité : quelles modalités de différenciation de l'espace ?

Depuis que les RMLL sont mobiles par projet – c'est-à-dire depuis 2004 – un comité sélectionne les villes hôtes pour l'année à venir et parfois la suivante. Il est composé de deux responsables volontaires de l'organisation (président, vice-président, trésorier, etc.) de chacune des quatre RMLL les plus récentes. Depuis juillet 2011, il faut y adjoindre un représentant de la dernière édition des RMLL décentralisées (RMLLd). Il compte donc au minimum huit puis neuf membres. Enfin, toute personne

1. Denis RETAILLÉ, « L'espace mobile », dans : *Le territoire est mort, vive les territoires ! : une (re)fabrication au nom du développement*, Paris : Institut de recherche pour le développement, 2005, p. 175–201.

2. Denis RETAILLÉ, *Les lieux de la mondialisation*, Paris : Le Cavalier Bleu, 2012, p. 20.

acceptée par ces derniers à la majorité des deux tiers peut l'intégrer. Ainsi, entre 2006 et 2010, l'Association Francophone des Utilisateurs de Linux (AFUL) y dispose d'un représentant. Ce comité, au temps de roulement beaucoup plus long que celui, annuel, des équipes d'organisation à proprement parler, assure le maintien de l'identité des RMLL à travers les années. En outre, il permet le transfert des compétences et l'accumulation de l'expérience.

La procédure de sélection est formalisée, dès avant 2007, dans un document qui reprend explicitement l'analyse multi-critères appliquée dans les grandes entreprises. Les critères, qu'ils soient destructifs (dont le non-respect est éliminatoire) ou sélectifs (dont le respect est valorisé), montrent que la sélection porte simultanément sur l'équipe d'organisation et sur le site d'accueil. Concernant le site, il est surtout question de la capacité d'hébergement et de restauration, mais aussi du nombre et de l'équipement des salles dans les structures d'accueil. La « situation géographique » est rapidement évoquée : c'est alors en fait de l'accessibilité du site en train et en avion, ainsi que de sa proximité au « centre-ville » dont il est question. Quant à l'organisation, elle doit être portée par « une ou plusieurs associations impliquées dans la communauté du Libre » d'au moins vingt personnes et compter au moins un membre de l'équipe d'une édition précédente.

Cette procédure est rendue nécessaire par la concurrence de plusieurs candidatures chaque année. On peut citer, à titre d'exemple, quelques candidatures malheureuses : Nice et Clermont-Ferrand (2007), Pau (2008), Bordeaux (2009), Toulouse (2010), Tarare (2012, face à Genève après le retrait de Liège, déjà sélectionnée, durant l'été 2011). Cependant, plus que la ville, c'est l'association qui est sélectionnée, à travers sa capacité à monter un dossier de candidature convainquant. En outre, les RMLL ne transforment pas le site sur lequel elles ont lieu, mais ont pour objectif de former les militants locaux à l'organisation d'événements et aux relations avec les collectivités territoriales – relations qui doivent tout de même exister un minimum auparavant dans la mesure où elles constituent un critère sélectif du dossier. La mobilité relève donc d'une stratégie de pénétration des territoires par le Libre où les bénévoles ne sont pas interchangeables, mais représentent au contraire des agents dont il s'agit d'activer les potentialités grâce aux compétences qui leur sont transmises

par le comité et des bénévoles d'années précédentes. Ainsi, pour François Pellegrini, membre du comité de 2005 à 2008 :

La mobilité des RMLL est un moyen de former nos activistes à l'organisation de manifestations, et au cours de l'organisation de l'événement, de prise de contacts avec le monde politique local et les médias locaux, et les acteurs associatifs locaux : les RMLL ont un rôle pédagogique crucial dans leur nature même sur la préparation de l'événement. Donc il est essentiel de « faire tourner le barnum », pour reprendre une expression qui avait été donnée par Thierry Laronde.[...] Aujourd'hui on a tous l'idée que faire tourner les RMLL c'est à chaque fois éduquer les politiques locaux et mettre au service des DSI de l'information gratuite [...]. Les RMLL laissent toujours des traces. Elles sont un outil essentiel de pénétration des territoires, donc il est essentiel de le faire tourner.

L'espace géographique est donc différencié par le comité en fonction de la capacité d'une association locale à transformer un site en lieu des RMLL. Dans ce cadre, la mobilité n'est pas un but en soi : elle est un moyen de s'ancrer dans les territoires. L'espace des représentations (la grammaire des représentations de l'espace) sous-tendu par les RMLL met ainsi en jeu des interactions entre mobilité et ancrage, mais aussi entre lieux et territoires.

Ainsi, la continuité de l'identité des RMLL est assurée par le comité. Elle leur confère une épaisseur temporelle qui laisse place à la différence, à l'évolution : elle leur donne une histoire. Or, le lieu n'est qu'une circonstance passagère.

1.1.3 Du lieu à la localité : innovations événementielles et acculturation

La théorie de l'espace mobile avance que les sites qui accumulent la mémoire des lieux qu'ils ont accueillis sont des localités¹. Une localité est donc le fruit d'une succession, sur un site, de lieux sédimentés en histoire. Les RMLL sont un lieu mobile avec une histoire. En trouvant leur site unique et les différents lieux qu'il accueille, elles apparaîtront comme une localité.

1. *ibid.*, p. 20.

Pour cela, chaque RMLL doit être envisagée comme un lieu spécifique. La dimension très discrète (par opposition à continue) de la manifestation, qui n'a lieu que quelques jours par an, autorise cela. De même, des innovations événementielles transforment pour certaines la portée et le sens des Rencontres. Elles montrent que les RMLL ont beaucoup évolué depuis leur première édition en 2000, qu'elles sont le fruit d'une construction de treize années et encore en cours. Elles sont aussi la preuve qu'une marge de manœuvre certaine est laissée aux organisateurs par le comité, que les associations locales possèdent une part de la maîtrise d'ouvrage. Parmi les innovations introduites, on peut compter : le village associatif (2003)¹, la parole aux enfants (2007, abandonnée après 2009)², les journées grand public³ – à différencier des conférences grand public – et le festival des arts numériques libres (2009)⁴, la vente de bières libres (2011), ou encore les *lightning-talks* (2012)⁵. L'organisation des RMLLd en 2011 est également à mettre au crédit de l'équipe strasbourgeoise, ainsi que des CÉMÉA (Centres d'Entraînement aux Méthodes d'Éducation Active) de La Réunion. Ces innovations constituent une acculturation de la manifestation de l'année n au contact de l'association locale qui organise celle de l'année $n + 1$. Plus important peut-être encore, la fréquentation de l'événement est multipliée par dix au cours de la décennie : en 2000, les RMLL rassemblent 500 personnes, contre 5500 en 2009. Cette croissance quantitative s'accompagne d'une transformation qualitative de l'organisation, des conférences et du public présent. Ainsi, le lieu mute tout en conservant son identité.

Or, seule la mémoire permet à l'identité de perdurer malgré le temps et l'altérité à soi-même qu'il induit. Le comité assure la mémoire des

1. Cela désigne le lieu constitué des stands des nombreuses associations du Libre qui se rendent aux RMLL : APRIL, Framasoft, Wikimedia France, etc. Leur liste est variable selon les années.

2. Il s'agit d'animations à destination des enfants réalisées soit en partenariat avec des structures de l'éducation populaire (2007 et 2009) soit avec des classes d'école primaire (2008). Les enfants sont invités à produire des contenus très variés à l'aide de logiciels libres auxquels ils sont initiés pour l'occasion.

3. Des journées sans conférence le week-end juste après (2009 et 2010) ou juste avant (2011 et 2012) le reste des RMLL. Le but est uniquement de présenter le Libre au grand public, à l'aide de stands, d'animations et d'ateliers.

4. Le terme désigne les concerts voire les projections d'œuvres libres qui ont lieu en marge des RMLL, notamment durant la nuit.

5. Littéralement conférences éclairs, elles ne durent pas plus de 5 minutes chacune.

RMLL : là où le comité se réunit (là où il a lieu) se trouve également ce site. Chaque année il se réunit physiquement aux RMLL, mais il se réunit aussi entre-temps grâce à divers outils aussi utilisés par les organisateurs et les participants : *mailing-lists* (listes de diffusion), *chan IRC* (salon de discussion instantanée – Relay Chat channels), wikis ou encore sites web. Autrement dit, si les RMLL ont une histoire, c'est qu'il faut avec Beaudé¹ considérer Internet comme un espace géographique de plein droit, où se trouvent des sites tout à fait capables d'accueillir des lieux et de devenir des localités.

1.2 D'un ancrage aquitain des RMLL à une image libriste de l'Aquitaine ?

Pour l'heure, revenons sur les innovations et acculturations que les RMLL ont connues, et dont la mobilité est la principale. En effet, lors de leur fondation les RMLL sont pensées pour être aquitaines, bordelaises même : elles sont ancrées dans un territoire. Cette origine aquitaine, connue des libristes présents aux Rencontres, demeure assez marquée aujourd'hui encore. Elle est d'une autre nature que la touche spécifique apportée chaque année par les équipes successives. Elle ne saurait pas plus être rapportée aux activités touristiques organisées lors des RMLL (visites du bassin d'Arcachon et du Médoc en 2000, du centre historique de Nancy et du musée de la brasserie en 2006, du château des ducs de Bretagne en 2009, ou encore du CERN en 2012). Aujourd'hui, l'Aquitaine est la seule région à disposer d'un stand fédérant ses différentes initiatives autour du Libre – voire du numérique en général : les pratiques observées lors des RMLL genevoises sont à rattacher à une politique de *regional branding* (assimilation de la région à une marque).

1.2.1 Dès l'origine, une volonté de promouvoir la région comme « territoire du Libre »

Il est encore possible de se référer à cet acte de naissance des RMLL qu'est le courriel de Pierre Jarillon de décembre 1999. Il y est prévu qu'elles rassemblent « les créateurs de logiciels libres ». Ce n'est qu'au

1. BEAUDE, *op. cit.*

printemps 2000, face aux demandes du Conseil Régional d'Aquitaine (CRA), que des thématiques grand public sont rajoutées. Le premier objectif des Rencontres est alors de « faire la promotion de la région et lui donner une image de technicité » : elles sont bien conçues comme un outil de promotion territoriale, de *regional branding*¹. Leur second objectif, quant à lui, est de « faire progresser les logiciels libres ». Par le lieu, le contact entre deux acteurs aux espaces des représentations différents est établi, visant à produire la codétermination d'une identité collective (les libristes sont chez eux en Aquitaine) et d'un territoire (l'Aquitaine favorise le logiciel libre). Il s'agit donc bien de construire, par la répétition de l'événement, un territoire du Libre², d'autant qu'initialement le CRA n'accepte de financer la manifestation qu'à la condition qu'elle devienne pérenne. Jean-Paul Chiron et Pierre Jarillon ont ainsi dû avancer 150 000 FF (22 500 €) afin d'engager les premières dépenses avant que la subvention ne soit débloquée. Les contraintes financières posées par le CRA ont donc, d'une certaine manière, encouragé la répétition voulue de l'événement.

Pourtant, dès 2003, les RMLL quittent Bordeaux pour Metz. Cette première mobilité n'est pas un projet : elle est une alternative proposée de l'extérieur à l'annulation des Rencontres cette année-là. Les organisateurs d'alors, dans leur récit, évoquent tous une grande fatigue à la suite de l'édition 2002 : « on n'arrivait pas à démarrer », « ça patinait », « on avait le sentiment de tourner en rond ». L'assemblée générale du 20 décembre 2002 décide, par conséquent, « l'annulation de la manifestation pour 2003 et la création d'un comité de pilotage afin de mieux préparer l'événement en 2004 »³. Le 21 janvier 2003, la décision est annoncée sur LinuxFr.org⁴. Peu après, un GULL lorrain propose de les organiser, idée débattue à l'ABUL. Pour certains, cette dernière devrait garder la maîtrise de l'événement : la mobilité est perçue comme un risque. Pour beaucoup d'autres en revanche, elle est une opportunité de se renouveler et de pro-

1. Gert-Jan HOSPERS, « Place marketing in Europe », dans : *Intereconomics* 39.5 (2004), p. 271–279.

2. Pierre-Amiel GIRAUD, *Les territoires du libre en Aquitaine*, Mémoire M2, Université Michel de Montaigne : Faculté de géographie, 2010, URL : <http://www.insolit.org/memoire-M2.pdf>.

3. Compte-rendu disponible sur : <http://www.abul.org/Compte-rendu-de-l-Assemblee,227.html>

4. <https://linuxfr.org/news/les-rencontres-mondiales-du-logiciel-libre-2003-nauront-pas-lie>.

mouvoir les logiciels libres auprès d'autres collectivités territoriales. La fréquentation par le grand public devient un enjeu, son importance devant démontrer aux institutions et aux politiques que le Libre est un sujet porteur.

1.2.2 L'Aquitaine et Bordeaux sont bien identifiés comme berceau des RMLL

Le rôle des collectivités territoriales, notamment du CRA et du Conseil Général de la Gironde (CG33) est unanimement souligné par les libristes bordelais : « sans la Région, rien n'aurait été possible », car elle soutient l'événement depuis le début. En 2010, les principales collectivités territoriales concernées (région, département, communauté urbaine) fournissent ensemble 120 000 € sur un budget total avoisinant les 280 000 €. Pour François Pellegrini, « tous les décideurs politiques de Bordeaux et d'Aquitaine connaissent les RMLL et savent que c'est là qu'elles sont nées ». Selon lui, elles ont un sentiment légitime de paternité des RMLL, qui permet de « tisser un lien affectif entre l'Aquitaine et le Libre ».

Cette paternité est aussi connue des participants aux RMLL, à l'exclusion peut-être du grand public. Une liste des villes hôtes la rappelle sur le site web de chaque édition et les discours la mentionnent souvent. Peut-être faut-il y voir également le fruit d'un engagement de certains individus dans les RMLL bien au-delà de l'organisation des éditions bordelaises. Ainsi, la délégation aquitaine est, en 2012 encore, très conséquente.

1.2.3 La mobilité des RMLL au service du *regional branding* numérique de l'Aquitaine

En 2012, trois anciens présidents des RMLL participent à l'animation du stand aquitain, situé dans le village des associations : Pierre Jarrillon (2000 à 2002), Jean-Paul Chiron (2004) et Jean-Christophe Élineau (2008). Ce stand est le seul à fédérer les initiatives d'un territoire. On y retrouve des informations sur les GULL régionaux, mais aussi sur ABULédu – distribution GNU/Linux destinée aux écoles et aux associations – ou encore sur les projets en cours d'Aquinetique, le pôle aquitain de compétences en informatique libre fondé à la suite du succès des RMLL montoises : Jean-Christophe Élineau en est d'ailleurs le président jusqu'à

fin 2011. La participation d'acteurs du Libre à de multiples associations de la mouvance (GULL, groupements professionnels, pôle de compétences, etc.) est très fréquente en Aquitaine. Elle est due à la faiblesse des effectifs à la fois impliqués et compétents. Au-delà du Libre, le stand présente des productions mettant en valeur les politiques numériques de l'Aquitaine, notamment des fascicules édités par le projet RAUDIN (Recherches Aquitaines sur les Usages pour le Développement des Dispositifs Numériques). Ce *regional branding* auprès de la mouvance du Libre s'appuie en outre sur une image de marque déjà bien assise : celle d'une région viticole. Ainsi, des dégustations de vin sont proposées sur le stand, à l'heure de l'apéritif.

1.3 Quelques ancrages plus ou moins implicites : le champ des mobilités possibles

Ayant présenté les RMLL comme un lieu mobile, il est toutefois aisé de relever qu'elles ont toujours eu lieu en France sauf cette année. Il y a donc, en sus de ceux définis explicitement par le comité, d'autres critères de choix qui, jusqu'à présent, ont disqualifié la grande majorité des sites. Ces critères peuvent même être intériorisés par les associations locales étrangères, qui n'envisagent pas la possibilité de candidater. Des représentations et des stratégies limitent donc le champ de mobilité des RMLL. Tout d'abord, les Rencontres sont un événement surtout francophone, ce qui pour plusieurs enquêtés est plutôt en contradiction avec leur prétention mondiale. Ensuite, la stratégie de pénétration territoriale explique en partie l'absence de Paris, ville mondiale de premier rang, parmi les hôtes des RMLL.

1.3.1 Les RMLL, un lieu du Libre francophone voire français ?

Les premières années, la dimension mondiale des RMLL ne fait pas débat : il s'agit de réunir le monde du Libre en un lieu pendant quelques jours. Les statistiques viennent à l'appui de cette idée : dès 2000, 31 nationalités sont représentées. Pourtant, aujourd'hui, la légitimité de l'objectif ne semble plus si assurée. Pour certains, ce n'est que cette année, en Suisse, que les RMLL sont véritablement devenues mondiales. Pour d'autres encore, cela n'est pas suffisant : « C'est un peu prétentieux de

se dire mondial quand on se balade entre Bordeaux, Strasbourg et Genève. Parce que bon, Genève c'est pas la France, mais ça reste tout près. Franchement, je pense qu'on pourra vraiment dire qu'elles sont mondiales quand elles auront lieu chaque année dans un pays différent. Hors d'Europe même. » La mobilité engendre donc un changement de sens de la mondialité. Caractérisée au début par la capacité des organisateurs à mobiliser des acteurs du Libre venant de l'ensemble de la planète, elle devient plutôt la capacité à avoir lieu en tout point du globe. En ne sélectionnant que des villes françaises (les seules à se porter candidates) les RMLL acquièrent l'image d'un événement français – certes déjà contenue dans le béret qu'arbore le manchot de la première affiche. Autrement dit, le territoire national devient l'espace de référence pour mesurer la mobilité. Un extrait du dossier nantais ¹ exprime bien cette idée :

Le parcours des RMLL a commencé à Bordeaux (au Sud-Ouest) pour s'aventurer dans l'Est (Dijon, Nancy), puis au Nord (Amiens) pour enfin revenir dans le Sud-Ouest (Mont-de-Marsan). Ce tour de France semble alors avoir oublié l'Ouest. C'est pourquoi Linux-Nantes et ses partenaires associatifs provenant de l'ensemble de la région se proposent d'accueillir cet événement.

La volonté de continuité entre les éditions limite le champ de la mobilité. Il y a beaucoup de bénévoles et de conférenciers réguliers : cette caravane des RMLL doit pouvoir se rendre sur place, et participer. En outre, pour que le transfert de compétences et d'expérience ait lieu, il faut que les membres du comité puissent s'impliquer dans l'organisation de l'événement. L'espace linguistique francophone est donc un premier ancrage. Le coût du transport en dessine un autre : les frais de voyage des conférenciers représentent déjà entre un tiers et la moitié du budget total des Rencontres, sans parler des participants venant sur leurs deniers, pour qui un coût élevé serait dissuasif.

En outre, les RMLL ont pour volonté d'attirer le grand public. Cela n'encourage pas les conférences dans d'autres langues que le français, malgré les initiatives du comité pour augmenter la proportion de communications en anglais. Développeurs et investisseurs étrangers peuvent

1. Disponible sur le site du comité des RMLL : <http://comite.rml.info/Dossiers-exemples.html>

donc être tentés de se diriger vers des lieux concurrents des RMLL d'avantage anglophones, quoique souvent à vocation plus spécialisée, tels que le FOSDEM (*Free and open source software developers' European meeting*) ou l'OWF (*Open World Forum*). Ce dernier, sorte d'équivalent des RMLL pour la mouvance *open*, a d'ailleurs choisi une stratégie spatiale toute différente. Ayant lieu chaque année à Paris depuis 2008, il illustre bien cet espace du marché qui vise l'accélération des flux (de la mobilité donc) par la fixation de lieux qui les contrôlent et les commutent.

1.3.2 La mise à distance de la métropole parisienne

À l'inverse, les RMLL n'ont jamais eu lieu à Paris. Certes, aucun GULL parisien n'a déposé de candidature, ce qui montre qu'il n'existe pas de volonté de leur part d'organiser la manifestation. Cependant, il s'agit aussi d'une stratégie du comité. Lorsque les *Ubuntu-parties* parisiennes peuvent drainer jusqu'à 5 000 personnes, autant que les RMLL, on pourrait penser que des Rencontres dans la capitale rencontreraient un franc succès auprès du grand public. Cependant, c'est oublier l'objectif de pénétration des territoires qu'ont les RMLL. Or, dans la métropole parisienne, les RMLL passeraient inaperçues, n'auraient aucun poids, d'autant que des événements concurrents y ont lieu, comme l'OWF ou Solutions Linux (10 000 visiteurs en 2010). Par ailleurs, d'autres acteurs du Libre se chargent déjà très efficacement de cette mission. L'Île-de-France est ainsi la première région européenne pour les emplois liés aux logiciels libres, et le pôle de compétitivité francilien System@tic comporte un groupe thématique Logiciels Libres. Enfin, les critères destructifs énumérés par le comité handicaperaient une éventuelle candidature de la capitale : y trouver une structure d'hébergement d'au moins 400 places à moins de 20 € la nuit relève de la gageure.

Au contraire, la modestie de la ville hôte est perçue comme un avantage. Jean Peyratout¹ affirme ainsi que « les retombées dépendent de l'implication préalable des acteurs locaux mais sont d'autant plus modestes que la ville est importante ». À cet égard, l'édition montoise est emblématique. Son succès est indéniable aux yeux des acteurs : c'est « un tour de force »,

1. Malory JESSAUME, *Les Rencontres Mondiales du Logiciel Libre : un outil de communication important pour le mouvement du logiciel libre*, Mémoire L3, Lyon : Université Lyon 2, 2012, p. 25.

une « réussite exemplaire ». « J'exagère à peine en disant que c'était l'événement de l'année à Mont-de-Marsan ». L'un ajoute : « on a augmenté la population de la ville de 10 % pendant une semaine, tout simplement ». L'exactitude de cette information est impossible à vérifier, mais le succès s'est concrétisé, entre autres, par la fondation du pôle de compétences Aquinetic, ainsi que par la publication du livret *Sur la route des logiciels libres* grâce à la participation du Conseil Général des Landes.

Le comité doit cependant composer avec les candidatures qui lui sont proposées. Mont-de-Marsan est ainsi la seule ville moyenne (aire urbaine dont la ville centre compte entre 20 000 et 100 000 habitants) à avoir accueilli les RMLL, alors qu'elle semble présenter le profil idéal de la ville hôte.

2. Un haut lieu du Libre, point d'entrée de la mouvance mondiale sur les territoires

Les sites successifs des RMLL dessinent donc en creux un ancrage territorial, faisant apparaître l'espace géographique simultanément homogène et différencié. Ce paradoxe se résout en rappelant qu'aux RMLL, plusieurs lieux sont en réalité présents sur un même site. Les libristes qui restent à discuter jusque tard dans la nuit des nouvelles fonctionnalités de leur logiciel ou des politiques numériques ne participent pas du même lieu que les participants qui viennent se former à l'imagerie et la visualisation médicale libres, ni même que le grand public qui vient se renseigner sur les enjeux sociétaux des logiciels libres. Cette deuxième partie analyse les Rencontres (donc les divise en plusieurs lieux) en fonction de leurs objectifs et des publics visés. D'abord, pour les militants du Libre, les RMLL sont un haut lieu, une célébration de leur monde vécu. Ensuite, elles sont un outil de sensibilisation des collectivités, des institutions et des politiques. Enfin, elles cherchent à inscrire le Libre dans des problèmes publics contemporains pour montrer que le logiciel libre est mis au service de la société.

2.1 Les RMLL comme conférence hacker, ou la fête comme catharsis

L'un des objectifs premiers des RMLL est de permettre la rencontre « des créateurs de logiciels libres » du monde entier. C'est pourquoi, dès le début, elles sont également appelées *Libre Software Meeting*, la notion de monde (*World*) disparaissant à la fois pour éviter l'assimilation à une multinationale et parce que la dimension mondiale est acquise hors de France du seul fait qu'elles ont lieu à l'étranger. À la fin des années 1990, deux environnements de bureau libres (KDE et GNOME) s'affrontent. Pour Pierre Jarillon, les causes du conflit ne sont au fond que des malentendus : une rencontre entre développeurs devrait les dissiper et produire des relations de confiance. Cette conviction, qu'il tire de son expérience professionnelle, peut être reformulée ainsi : la proximité spatiale des acteurs est par elle-même, et indépendamment du contexte extérieur, génératrice d'aménités (ici la confiance). Il s'agit de révéler ce qui rassemble, et non ce qui sépare. Les RMLL sont donc une fête célébrant un monde vécu et permettant, le reste de l'année, une saine collaboration entre les projets : les RMLL sont un lieu hybride, terme qu'il faudra expliquer.

2.1.1 Les Rencontres du Monde du Logiciel Libre : la célébration d'un monde vécu

Lors des entretiens, certains affirment qu'il « fallait qu'on puisse manger, boire, pisser ensemble.[...] Ça améliore beaucoup les relations. » Cette approche rejoint celle de Di Méo¹ :

la fête agit [...] comme une véritable catharsis. Le plus souvent, elle désamorçait les conflits dans une sorte de rituel. Au-delà des disputes, des inégalités, des injustices, des luttes et des clivages sociaux, spatiaux, religieux ou politiques, ce rituel indique que l'unité du groupe finit toujours par l'emporter. Cette unité s'impose en tant que valeur essentielle et existentielle, autant que nécessité profonde de survie territoriale autant que sociale.

1. Guy DI MÉO, *La géographie en fêtes*, Paris-Gap : Géophrys, 2001, p. 17.

Le terme de rituel ne paraît pas excessif, d'autant qu'à chaque édition a lieu un « Repas du Libre » qui réunit principalement les « militants du code »¹. Or, pour Gomez² :

d'une manière générale, le repas est un facteur, sauf cas pathologique, de resserrement des liens entre les individus. [...] La nourriture est chose de partage, elle joue un rôle irremplaçable pour forger une relation durable. On ne saurait la partager de manière neutre fût-ce avec un inconnu.

Le repas manifeste l'unité du monde du Libre : l'échelle convoquée, consolidée, est bien celle mondiale. Pourtant, cette consolidation ne peut se faire que par le passage à une échelle où l'espace ne peut plus intervenir comme boucle de rétroaction négative aux relations sociales. Depuis 2011, cette boisson symbolique qu'est la bière libre (dont la recette et le procédé de fabrication sont publics) joue un rôle similaire³. La fête, en abolissant toutes les distances, fait figure de lieu total : elle est une célébration, parfois jusqu'à la religiosité. Deux participants m'ont ainsi affirmé venir aux RMLL « en pèlerinage » afin de « croire encore qu'il y a un espoir pour le Libre »⁴.

Coleman⁵ montre bien comment les DebConf sont la célébration d'un monde vécu (« the celebration of a lifeworld »). Elle insiste ainsi sur la manière dont elles sont la mise en scène ritualisée de pratiques et de représentations courantes dans la vie quotidienne des membres, portées à un très fort degré d'intensité par la présence physique des membres. Or,

1. Serge PROULX, « Les militants du code : la construction d'une culture technique alternative », dans : *Le logiciel libre en tant que modèle d'innovation sociotechnique. Pratiques de développement et de coopération dans les communautés*, Montréal : Université McGill, 2006.

2. Florent GOMEZ, *Le repas au collège : aspects psychosociologiques de la commensalité scolaire*, Talence : Presses universitaires de Bordeaux, 1985.

3. Cela est d'autant plus vrai que pour différencier en anglais libre et gratuit (qui se disent tous deux *free*), Richard Stallman dit souvent « think of *free* as in free speech, not as in *free beer* ». Dans son texte sur la définition du logiciel libre, cette phrase de clarification vient même avant la liste des quatre libertés. Il s'agit donc pour les libristes de jouer avec – de hacker ? – leurs propres référents culturels. Voir Richard M. STALLMAN, *Free Software, Free Society : Selected Essays of Richard M. Stallman*, Boston : GNU Press, 2002, p. 43.

4. Là encore il faut faire référence à un contexte culturel plus large, que nous n'avons pas le temps de développer ici. Pour plus de précisions sur le sujet : Christopher KELTY, *Two bits : the cultural significance of free software*, Durham : Duke University Press, 2008, p. 64-94 ; GIRAUD, *op. cit.*, p. 85-94.

5. COLEMAN, *op. cit.*

les deux premières DebConf ont lieu à Bordeaux (DebConf 0 et DebConf 1), dans le cadre des RMLL 2000 et 2001. Elles sont ensuite devenues complètement indépendantes, et sont uniquement une conférence *hacker* – ici, passionnés des techniques de l’informatique (« *aficionados dedicated to the craft of computing* »). Elles démontrent cependant qu’une telle conférence est au fondement de l’un des lieux des RMLL. Les Rencontres sont ainsi une technique spatiale de « mobilisation des communautés distantes »¹, un espace-temps extraordinaire pour favoriser la collaboration ordinaire.

2.1.2 Entre territorial et réticulaire, les RMLL sont un lieu hybride

Pour comprendre les RMLL, dans leur fonctionnement comme dans leurs enjeux, il est nécessaire d’envisager Internet comme un espace géographique. Pour cela, le concept de lieu doit être affiné, en distinguant lieux territoriaux et lieux réticulaires. Les lieux territoriaux mettent en œuvre des moyens d’abolition de la distance par contiguïté. Le contact, physique, y stimule tous les sens. Avec les lieux réticulaires, au contraire, « la non-pertinence de la distance est fondée sur la connexité »². Ce concept permet aux géographes d’investiguer cet espace qu’est Internet. Les libristes fréquentent ce dernier avec assiduité y compris durant les Rencontres, au point que le maintien du réseau en état de marche est un défi majeur pour les organisateurs. Les RMLL ont aussi lieu sur Internet, les lieux réticulaires étant d’ailleurs les seuls sur lesquels la manifestation existe entre deux éditions. Pourtant, par leurs pratiques, les participants saisissent du même geste connexe et contigu, discutant par exemple sur le *chan IRC* alors qu’ils sont dans la même pièce. Les RMLL sont donc un lieu hybride, mêlant le territorial au réticulaire de manière si intense que l’artificialité de la distinction (pourtant nécessaire à l’analyse) ressort.

L’hybridation de l’espace est générale : elle touche l’ensemble de la société, et pas seulement la mouvance du Libre³. Pourtant, dans le cas des RMLL, elle revêt un sens particulier. Les libristes envisagent le numérique

1. JULLIEN, DEMAZIÈRE et HORN, *op. cit.*

2. Boris BEAUDE, « Éléments pour une géographie du lieu réticulaire », thèse de doct., Paris : Université Paris 1, 2008, p. 191.

3. *idem*, *Internet : changer l’espace, changer la société : les logiques contemporaines de synchronisation*, p. 209.

et Internet de manière spécifique, ce qui rend l'hybridation elle-même originale. La manifestation de cette hybridité est au cœur de la catharsis que permettent les RMLL : partager un monde vécu, c'est avant tout partager ses métriques (manières dont on mesure la distance).

2.2 Les RMLL comme outil de sensibilisation et de formation des collectivités et des institutions

Un autre objectif des RMLL est de promouvoir l'usage et la production de logiciels libres. Dans le cadre de l'organisation des RMLL, les institutions sont facilement approchées. En outre, elles partagent le fort intérêt des activistes pour l'idée de mutualisation des moyens, « accélérateur » de leur adoption des logiciels libres¹. Elles sont donc perçues comme des clés d'entrée pour faire pénétrer le Libre dans les territoires. Or, les représentants locaux de la mouvance sont justement les organisateurs des RMLL : faire pénétrer le Libre revient à augmenter leur influence auprès des décideurs.

2.2.1 Les collectivités sont perçues comme des acteurs clés pour la pénétration des territoires

Les collectivités, par les politiques qu'elles mènent et par les rapports fréquents qu'elles entretiennent avec leurs administrés, ont un fort pouvoir de prescription, y compris en matière de logiciel. Par exemple, il faut pouvoir ouvrir facilement les documents disponibles sur les sites institutionnels. Souvent, les quelques conseils prodigués pour y parvenir sont suivis. C'est pourquoi la *Free Software Foundation Europe* (FSFE) vise surtout ce genre de sites web dans le cadre de sa campagne pour la promotion des lecteurs PDF libres. La sensibilisation des acteurs publics est donc perçue comme un moyen d'atteindre *in fine* le grand public et les entreprises. À ce titre, les Chambres de commerce et d'industrie (CCI) sont des organismes souvent prisés par les organisateurs : la CCI locale est partenaire au moins des éditions 2006, 2007, 2009 et 2011.

Par ailleurs, pour les organisateurs comme pour de nombreux militants, la conversion des institutions au Libre est une cause citoyenne. Si, pour

1. François ÉLIE, *Économie du logiciel libre*, Paris : Eyrolles, 2009.

eux, elle peut donc être un but en soi, elle est surtout présentée comme un moyen, par exemple pour réduire les dépenses et s'assurer que « l'argent public ne paie qu'une fois »¹. Il faut donc clairement différencier les temps de sensibilisation et de formation. La première est réalisée en amont des RMLL, par les organisateurs. Elle doit être efficace car elle doit aboutir à l'octroi de subventions : on comprend pourquoi les opérations de transfert d'expérience et de compétence débutent de nombreux mois à l'avance, parfois plus de douze. D'une certaine manière, la sensibilisation se poursuit pendant les RMLL, la rencontre avec d'autres militants pouvant apporter de nouveaux éléments. En outre, la fréquentation par le grand public peut encourager à envisager la question comme un enjeu électoral, même secondaire. Pourtant, les RMLL sont plutôt le temps de la formation. C'est d'ailleurs ainsi que les organisateurs présentent les choses : un premier retour sur investissement est fait par les partenaires publics dès lors qu'ils bénéficient durant l'événement de formations les intéressant directement. Il y a donc toujours un ou deux thèmes dont l'intitulé tourne autour des termes administrations, collectivités territoriales et politiques publiques, soit au total plus d'une dizaine de conférences par an.

Cette stratégie visant, à l'échelon régional voire départemental ou communal, une diffusion *top-down* des logiciels libres est souvent couronnée de succès, surtout dans le cas de Mont-de-Marsan. La préface du livret *Sur la route des logiciels libres* est ainsi signée par Henri Emmanuelli, président du Conseil général des Landes. De même, la participation d'hommes politiques de premier plan montre que les RMLL ont su imposer les logiciels libres comme un thème (mineur) de la vie politique française, et s'imposer comme le lieu légitime depuis lequel en parler. C'est ainsi qu'en 2006, à Vandœuvre-lès-Nancy, une table ronde sur les brevets logiciels et la loi DADVSI a réuni François Bayrou (UDF) et Michel Rocard (PS), mais aussi Richard Cazenave (UMP) et Martine Billard (Verts).

1. *ibid.*, p. 40.

2.2.2 Un levier pour la reconnaissance et l'influence des associations locales ?

Les cas de Mont-de-Marsan et d'Aquinetic ne sont pas isolés. Un autre exemple possible est celui du rôle qu'ont pu jouer certains membres de l'ABUL dans la lutte contre les brevets logiciels grâce aux RMLL.

Entre 2000 et 2002, des membres de l'ABUL acquièrent les compétences nécessaires au montage d'événements de grande ampleur, notamment au vu des moyens humains dont ils disposent. Sur les questions numériques, l'influence et la crédibilité de l'association auprès des hommes politiques locaux croissent rapidement. L'exemple le plus frappant remonte peut-être à la première édition. En 2000, les instances européennes réfléchissent à l'évolution des brevets en Europe : le 5 juillet, soit le premier jour des RMLL, la Commission Européenne propose la création d'un brevet communautaire. De même, une conférence intergouvernementale est prévue pour novembre afin de réviser la Convention sur le Brevet Européen (CBE) de 1973 et d'introduire la brevetabilité des logiciels¹. Dans ce contexte, la conférence de Jean-Paul Smets sur ce dernier sujet est très suivie. Notamment, Gilles Savary, alors député européen et conseiller municipal de Bordeaux, est convaincu par les arguments des libristes : lors de son discours de clôture, il s'oppose publiquement aux brevets logiciels et conseille d'envoyer des personnes faire du lobbying au Parlement européen :

En tant que député européen, je me mets à votre disposition pour que vous fassiez entendre votre voix auprès des institutions européennes, et je vous invite et vous propose de venir dans les quatre ans qu'il me reste au parlement européen rencontrer des députés pour les sensibiliser à ces questions de la protection de la création et notamment de la création des logiciels.

Le Parlement européen est un jeune parlement qui accueille volontiers les lobbies et vous ne devez pas laisser la place aux seuls lobbies des firmes et des industriels.²

1. Antoine BILLET, « Brevets logiciels : mode d'emploi », dans : *OINet* (2000), URL : <http://www.01net.com/editorial/111405/brevets-logiciels-mode-demploi/>.

2. Transcription du discours de Gilles Savary : <https://www.april.org/articles/communiqués/pr-lsm.html>.

François Pellegrini y est envoyé. Jusqu'à la demi-victoire des opposants aux brevets logiciels (aucun texte n'est finalement voté), il est l'un des conseillers de Michel Rocard sur cette question.

Ce succès spectaculaire des RMLL pour accroître l'influence des libristes auprès des hommes politiques locaux (qui ici ont ensuite servi d'intermédiaires vers d'autres hommes politiques) n'est pas une exception. À différents niveaux, sur des sujets différents, plusieurs RMLL ont permis d'asseoir la notoriété des associations organisatrices, de densifier leurs réseaux de relations, de faire germer des projets. Souvent, elles bénéficient aussi, avant ou pendant la manifestation, d'une couverture accrue dans la presse quotidienne régionale (*Sud-Ouest* pour les éditions bordelaises et montoises, *20 minutes Strasbourg* en 2011, *L'Express* pour Genève). La venue des RMLL permet donc aux associations locales d'acquérir temporairement une visibilité pour faire valoir, sur l'aire (régionale) de diffusion du journal, une cause pour elles universelle.

Cependant, l'organisation d'une telle manifestation est harassante, et les bénévoles les plus investis peuvent en sortir exsangues. Certains GULL organisateurs ont eu du mal à s'en remettre. C'est pourquoi, par la suite, les partenariats entre associations multiples sont favorisés – cela permet en outre de resserrer les liens entre lesdites associations. En 2009 par exemple, Linux-Nantes s'associe avec Alliance Libre : le GULL bénéficie des contacts du groupement professionnel. Dans ce cas, la diversité de la mouvance apparaît comme une ressource importante dans la production du lieu des RMLL.

2.3 Une manifestation inscrite dans le monde : la prédication auprès du grand public

Enfin, le grand public est visé pour lui-même. Les RMLL sont un haut lieu du Libre francophone, perçu comme tel par les participants. Cependant, il est aussi une interface avec un grand public à sensibiliser voire convertir. Pour prendre une image, les RMLL sont à la fois un lieu de pèlerinage et une terre de mission. Le premier contact est assuré par la dimension festive de l'événement, d'autant plus lorsqu'elle est inscrite dans l'espace public. Il peut être prolongé par des conférences qui visent à intégrer la mouvance du Libre aux grands problèmes publics contemporains.

2.3.1 La fête comme stratégie de conquête territoriale

La fête, déjà évoquée plus haut comme célébration d'un monde vécu entre les militants, est aussi un vecteur de conquête territoriale auprès du grand public. En 2010 par exemple, des présentations sont organisées dans des cinémas. Le festival des arts numériques libres relève aussi de cette stratégie : il s'agit d'utiliser la fête pour produire du contact (donc du lieu) avec le grand public et en profiter pour essayer de le sensibiliser voire de le faire venir à des conférences accessibles sans bagage technique. Dès lors, la localisation fine des sites au sein de la ville hôte est très importante, tout comme l'occupation de l'espace public en amont de l'événement à l'aide d'affiches ou de manifestations en rapport avec le Libre. C'est ainsi que l'on peut interpréter l'affluence bien plus faible du grand public à Genève qu'à Bordeaux. En 2010, de nombreuses affiches sont disposées dans la CUB, surtout à Bordeaux même et Talence, plusieurs semaines à l'avance. Le week-end grand public est organisé sur les quais, à proximité du skate-parc, c'est-à-dire un lieu fréquent de pause sur l'un des parcours de promenade préférés des Bordelais. À Genève, s'il est vrai que des publicités dans les tramways annoncent la tenue des RMLL, l'occupation de l'espace public est beaucoup plus discrète, pour ne pas dire invisible. La salle communale de Plainpalais, où se déroule le week-end grand public, semble à l'écart des lieux de loisirs genevois.

Cela change le sens de l'événement, en faisant varier l'importance de l'une de ses dimensions. À Bordeaux, les RMLL sont inscrites dans la vie des festivals, jouent leur rôle d'interface « entre les geeks et toutes les variétés de non-geeks », pour reprendre une formule de François Pellegrini. À Genève, l'édition 2012 est davantage hors-sol, sans interaction véritable avec le grand public. Cela n'enlève rien au succès des deux dimensions des RMLL évoquées plus haut. Plusieurs participants évoquent même le très bon niveau des conférences et soulignent la qualité de l'organisation.

2.3.2 L'inscription du Libre dans de grandes questions de société

Les thèmes abordés constituent l'autre élément de la stratégie des RMLL à destination du grand public. Les organisateurs veulent montrer que le Libre est important au-delà du logiciel : qu'il est porteur d'un modèle de société libre, ou au moins qu'il vise à mettre les logiciels au service

de la société. Des thèmes choisis chaque année, transversaux ou non, répondent à cette exigence. Par exemple, l'édition 2000 traite la question du développement (économique). L'espace visé est surtout l'Afrique. L'un des résultats de ces premières RMLL est d'ailleurs la création de l'Association Africaine des Utilisateurs de Logiciels Libres (AAUL). Les questions d'éducation (populaire ou non), la santé, l'économie, la culture ou encore le développement durable et l'accessibilité (thèmes transversaux en 2010) dénotent ainsi une volonté d'inscrire le Libre dans les problèmes publics contemporains.

Cela montre que, pour les militants, les logiciels – *a fortiori* ceux qui sont libres – ne sont pas des outils neutres, qu'ils sont porteurs d'un sens politique et d'une manière d'être au monde. Il s'agit, autrement dit, de les introduire eux-mêmes comme problème public – au sens où le *probouleuma* désigne dans la *polis* athénienne une proposition présentée au Conseil, et qui par conséquent fait débat¹. L'hybridation de l'espace géographique contemporain rend cette introduction d'autant plus nécessaire.

2.3.3 Les limites des RMLL : horizon, ou confins ?

Plusieurs lieux se superposent au cours des RMLL : le lieu de la conférence technique, celui de la formation et de la sensibilisation des collectivités, enfin celui de la rencontre grand public. Ces trois lieux, qui sont en interaction, ne relèvent pourtant pas du même espace de représentation, dans la mesure où c'est justement cet espace de représentation commun que les RMLL visent à produire. Vers l'intérieur, la dimension spatiale de ces lieux est identique, dans la mesure où la distance n'y est pas pertinente. En revanche, ils se différencient par la relation qu'ils entretiennent avec l'espace qui les englobe, et donc par la forme de leur limite².

D'abord, en tant que conférence *hacker*, les Rencontres ont la prétention de manifester un monde du Libre (comme le laissait entendre un nom un temps avancé pour les RMLL : *Libre World*) qui recouvre l'intégralité du Monde. Un tel lieu peut être tout à fait mobile, à l'image de son *spin-off* que sont les DebConf. Le site qui les héberge et la localité à laquelle il

1. NdE : *probouleuma*, équivalent d'une proposition de loi présentée devant la *Boulè*, conseil de citoyens – initialement tirés au sort – en charge du gouvernement. Ce sont les premières formes de démocratie politique.

2. RETAILLÉ, « La transformation des formes de la limite ».

participe est indifférent à ce qui se passe à l'intérieur du lieu. La limite du lieu se dérobe donc toujours, aussi bien pour les participants que pour l'observateur : c'est un *horizon* (première forme de la limite). On est donc toujours à l'intérieur du lieu qui se manifeste (le monde du Libre), ce qui signifie aussi que même en n'étant pas sur site on participe au lieu.

Ensuite, en tant qu'outil de prédication et de pénétration des territoires, les RMLL apparaissent comme des missions (dans le sens par exemple des missions jésuites d'Amérique du Sud) temporaires. La différence de valeur entre l'intérieur (qui justement porte les valeurs à promouvoir) et l'extérieur est postulée. Le territoire d'accueil y est vu comme un espace à « libérer » dont il s'agit de changer la nature pour en faire un territoire du Libre – c'est en ce sens que la conversion de collectivités au Libre peut relever de la catastrophe (au sens de René Thom : transformation soudaine et profonde) territoriale¹. La limite de ce lieu prend donc la forme des *confins* (deuxième forme).

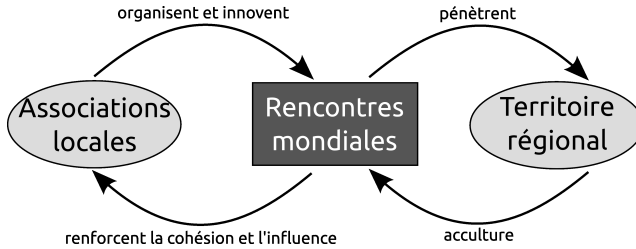
Enfin, les RMLL adoptent en partie les *frontières* (troisième forme) des collectivités territoriales partenaires, à la fois à travers les formations destinées à leurs personnels, la part importante qu'elles jouent dans le financement de l'événement et car elles sont perçues comme un acteur clé pour la « libération » des territoires. Les collectivités territoriales ont bien une limite repérable, mais à la différence de ce qui se passe pour les confins, elles sont toutes de même nature (ce qui explique qu'on puisse les juxtaposer si elles relèvent du même échelon, ou bien les emboîter si elles sont de niveau différent).

Ces trois lieux se manifestent simultanément et mobilisent pour une part les mêmes acteurs, qui n'y voient rien à redire. C'est que si plusieurs lieux peuvent se produire au même moment sur le même site, il est aussi possible d'être en même temps dans tous ces lieux.

3. Conclusion : les RMLL, lieu privilégié d'observation de la mouvance

Pour synthétiser, les RMLL sont un haut lieu mobile du Libre francophone qui vise, à travers l'organisation d'une double rencontre entre li-

1. GIRAUD, *op. cit.*, p. 36.



Jeux d'échelons et d'interface au lieu des RMLL

bristes et entre les libristes et le grand public, à pénétrer les territoires régionaux. Elles confèrent ainsi une légitimité aux associations locales face aux partenaires institutionnels.

Nous avons montré les co-constructions des représentations, stratégies et espaces (lieux comme territoires) à l'œuvre aux RMLL. Les différents lieux qui y sont co-présents interagissent pour former les RMLL, répondant chacun à un objectif déterminé. Depuis le comité qui assure la mémoire des RMLL et le transfert de compétences jusqu'au grand public à conquérir en passant par la célébration d'un monde vécu, tout concourt à montrer que les libristes possèdent un espace des représentations hybride, capable de mettre à profit les potentialités relationnelles des espaces géographiques indépendamment de leurs métriques. Bien que l'hybridation soit aujourd'hui en voie de généralisation, les militants du code sont parmi les seuls à refuser de laisser Internet hors du champ du politique. L'observation de leurs pratiques permet de déceler des signaux faibles mais aussi, par contraste, de pointer des dangers d'une manière de n'aborder le numérique que par ses littoraux visibles, laissant l'essentiel du continent à l'état de *terra incognita* à la merci d'acteurs dont les intérêts ne sont pas forcément ceux des citoyens.

Les évolutions des RMLL témoignent aussi des transformations du Libre au cours de la décennie passée. Le profil des associations organisatrices change, reflétant l'acculturation du Libre dans la société mais aussi la perte relative d'importance des GULL dans le mouvement associatif du Libre (puisque'ils n'en sont plus les seuls acteurs). Ainsi, jusqu'en 2007, les organisateurs ne sont que des GULL. De 2008 à 2010, des GULL sont porteurs, épaulés par d'autres associations de la mouvance. En 2011

et 2012, les associations porteuses ne sont plus des GULL : ce sont eux qui désormais épaulent.

Finalement, d'abord centrées exclusivement sur les logiciels, les Rencontres se sont élargies jusqu'à inclure des productions culturelles, du matériel et même des données. Cet élargissement du champ du Libre, qui avec des pratiques amène des cultures et des mondes vécus nouveaux, questionne sans cesse ce qui fait l'unité de la mouvance, alors même que son orientation générale semble conservée. C'est donc à juste titre que Kelty¹ a pu qualifier les mouvements dérivés de celui du logiciel libre de « modulations ». Les opportunités en sont l'enrichissement et la fertilisation croisée, les risques en sont la disparition d'un sens commun même minimal du Libre ainsi qu'une perte de lisibilité de la mouvance, déjà complexe pour le grand public. Quelle image du Libre un visiteur peut-il avoir quand il vient d'assister, aux RMLL, à une conférence présentée à l'aide d'un programme propriétaire ? En se modulant, le Libre a pu accroître sa diffusion auprès de divers milieux professionnels et culturels, ainsi que d'une partie du grand public. Pourtant, son succès risque de se faire au détriment des valeurs et des pratiques des développeurs qui ont initié le mouvement. Ce n'est pas un échec des RMLL : elles ne font que rendre visible, en la condensant, la mouvance du Libre dans toute sa diversité. L'absence de certitude concernant les pratiques et les valeurs constitue justement l'un des indices de l'espace mobile et justifie le terme même de mouvance : on ne peut jamais dire si l'on est dehors, ou bien si l'on est dedans.

Références

BEAUDE, Boris, *Internet : changer l'espace, changer la société : les logiques contemporaines de synchronisation*, Limoges : Fyp éditions, 2012.

– « Éléments pour une géographie du lieu réticulaire », thèse de doct., Paris : Université Paris 1, 2008.

BERQUE, Augustin, « Lieu », dans : *Dictionnaire de la géographie et de l'espace des sociétés*, sous la dir. de Jacques LÉVY et Michel LUS-SAULT, Paris : Belin, 2003, p. 555–556.

1. KELTY, *op. cit.*

- BILLET, Antoine, « Brevets logiciels : mode d'emploi », dans : *01Net* (2000), URL : <http://www.01net.com/editorial/111405/brevets-logiciels-mode-demploi/>.
- COLEMAN, Gabriella, « The Hacker Conference : A Ritual Condensation and Celebration of a Lifeworld », dans : *Anthropological Quarterly* 83.1 (2010), p. 47–72.
- DI MÉO, Guy, *La géographie en fêtes*, Paris-Gap : Géophrys, 2001.
- ÉLIE, François, *Économie du logiciel libre*, Paris : Eyrolles, 2009.
- GIRAUD, Pierre-Amiel, *Les territoires du libre en Aquitaine*, Mémoire M2, Université Michel de Montaigne : Faculté de géographie, 2010, URL : <http://www.insolit.org/memoire-M2.pdf>.
- GOMEZ, Florent, *Le repas au collège : aspects psychosociologiques de la commensalité scolaire*, Talence : Presses universitaires de Bordeaux, 1985.
- HOSPERS, Gert-Jan, « Place marketing in Europe », dans : *Intereconomics* 39.5 (2004), p. 271–279.
- JESSAUME, Malory, *Les Rencontres Mondiales du Logiciel Libre : un outil de communication important pour le mouvement du logiciel libre*, Mémoire L3, Lyon : Université Lyon 2, 2012.
- JULLIEN, Nicolas, Didier DEMAZIÈRE et François HORN, « How free software developers work. The mobilization of *distant communities* », dans : *Cahier de recherches* 7 (2006), URL : http://www.marsouin.org/IMG/pdf/Demaziere-Horn-Jullien_7-2006.pdf.
- KELTY, Christopher, *Two bits : the cultural significance of free software*, Durham : Duke University Press, 2008.
- LASSERE, Frédéric, « Internet : la fin de la géographie ? », dans : *Cybergeo : European Journal of Geography* (2000), URL : <http://cybergeo.revues.org/4467>.
- O'BRIEN, Richard, *Global financial integration : the end of geography*, London : Royal Institute of International Affairs ; Chatham House papers, 1992.
- PROULX, Serge, « Les militants du code : la construction d'une culture technique alternative », dans : *Le logiciel libre en tant que modèle d'innovation sociotechnique. Pratiques de développement et de coopération dans les communautés*, Montréal : Université McGill, 2006.

- RETAILLÉ, Denis, « La transformation des formes de la limite », dans : *Articulo. Journal of Urban Research* 6 (2011), URL : <http://articulo.revues.org/1723>.
- *Les lieux de la mondialisation*, Paris : Le Cavalier Bleu, 2012.
- « L'espace mobile », dans : *Le territoire est mort, vive les territoires ! : une (re)fabrication au nom du développement*, Paris : Institut de recherche pour le développement, 2005, p. 175–201.
- SHEPPARD, Elizabeth, « Problème public », dans : *Dictionnaire des politiques publiques*, Paris : Presses de Sciences Po., 2010, p. 530–538.
- STALLMAN, Richard M., *Free Software, Free Society : Selected Essays of Richard M. Stallman*, Boston : GNU Press, 2002.
- TURNER, F., « How Digital Technology Found Utopian Ideology : Lessons From the First Hackers' Conference », dans : *Critical Cyberculture Studies : Current Terrain, Future Directions*, 2006.

III

Trajectoires du Libre

Influence du Libre dans l'histoire du jeu vidéo

Damien DJAOUTI

À première vue, la définition du logiciel libre semble en tout point contraire à celle des jeux vidéo disponibles dans le commerce. En effet, pour reprendre les termes de la Free Software Foundation (FSF) : « L'expression « logiciel libre » veut dire que le logiciel respecte la liberté de l'utilisateur et de la communauté. En gros, *les utilisateurs ont la liberté d'exécuter, de copier, de distribuer, d'étudier, de modifier et d'améliorer le logiciel* »¹. Or, dans le cas des jeux vidéo, il s'agit de logiciels dont le code source n'est pas accessible, empêchant ainsi toute initiative d'étude, de modification ou d'amélioration. De même, leur nature commerciale empêche aussi leur copie et leur libre distribution. Enfin, du fait d'un effet de bord issu des mesures techniques de protection contre la copie², la liberté d'exécution de la plupart des jeux vidéo commerciaux actuels est également de plus en plus restreinte. Sur ordinateur, les jeux vidéo achetés dans le commerce ne peuvent être installés que sur un nombre limité de machines différentes ; tandis que le secteur des consoles de jeux vidéo

1. Relevé le 16-07-12 à partir de <http://www.gnu.org/philosophy/free-sw.fr.html>.

2. Les systèmes de type Digital Rights Management (DRM) sont des dispositifs techniques limitant, par exemple, le nombre d'installations possibles d'un jeu vidéo à partir de chaque exemplaire acheté, ou encore empêchant l'exécution d'un programme de jeu si l'utilisateur n'est pas connecté à Internet afin de valider l'unicité de son exemplaire. Pour plus de détails, voir la page Wikipédia http://fr.wikipedia.org/wiki/Gestion_des_droits_numériques.

oblige parfois l'utilisateur à posséder une machine spécifique s'il souhaite jouer à certains titres populaires¹.

Pourtant cette industrie, dont le modèle économique repose sur une logique « propriétaire » doit, indirectement, sa naissance au logiciel libre, ou tout du moins à sa philosophie. À travers ce chapitre, nous proposons de revenir aux prémices de l'histoire du jeu vidéo. Si ses débuts commerciaux s'accompagnent ensuite de l'arrivée massive du logiciel propriétaire, l'influence du logiciel libre ne disparaît pas pour autant. Nous aborderons également deux autres influences notables du Libre dans l'histoire du jeu vidéo. Tout d'abord, la pratique du *modding*, qui consiste à modifier des jeux existants ; puis la création de jeux vidéo en amateur, qui culmine aujourd'hui à travers le *Jeu 2.0*. L'étude de ces trois aspects de l'histoire du jeu vidéo nous permettra d'analyser les différents types d'influence du Libre sur le jeu vidéo.

1. Le Libre comme philosophie originelle des jeux vidéo ?

1.1 Les premiers jeux vidéo

Les tout premiers jeux vidéo recensés à ce jour sont le résultat de travaux de chercheurs travaillant dans des laboratoires universitaires britanniques ou américains².

Par exemple, le premier jeu sur ordinateur connu à ce jour pour utiliser un affichage sur écran vidéo est *OXO*. Créé en 1952 par Alexander Shafto Douglas, il s'agit d'un jeu de morpion tournant sur l'ordinateur EDSAC. La motivation première de Douglas pour la création de ce jeu vidéo était la recherche scientifique, *OXO* venant illustrer son mémoire universitaire sur l'Interaction Homme-Machine (IHM). En effet, si son principe est enfantin, *OXO* constitue un véritable tour de force technique. La particularité de l'EDSAC provient de sa capacité à stocker des programmes en mémoire (comme la RAM de nos ordinateurs actuels). L'appareil possède ainsi trois écrans CRT qui affichent l'état courant de la mémoire sous forme graphique. Le génie de Douglas a été de détourner cette fonctionnalité de

1. Politique d'exclusivité de certains titres très populaires (*Mario*, *God of War*, *Zelda*, *Gran Turismo*, *Halo*...) sur une console donnée afin d'augmenter ses ventes.

2. Damien DJAOUTI, « Les pionniers du jeu vidéo », dans : *Pix'n Love* 11 (2010).

contrôle de la mémoire pour en faire un moyen de synthèse graphique. Un écran CRT de contrôle, d'une résolution de 35x16, est programmé pour afficher une grille de morpion ainsi que les signes déposés par les joueurs, permettant donc de visualiser l'état de la partie. De plus, le panneau de contrôle de l'ordinateur est équipé un téléphone à cadran rotatif. Le chercheur l'utilise comme manette de jeu. Ce cadran comporte des numéros allant de 0 à 9, tandis qu'une grille de morpion est composée de 9 cases. On numérote donc les cases de la grille de 1 à 9 en commençant par celle située en haut à gauche. Pour jouer, l'utilisateur doit alors composer un numéro sur le téléphone afin d'indiquer dans quelle case il veut déposer son signe. Comme la plupart des projets des pionniers de l'informatique, le code source d'*OXO* est accessible sans restriction. Pour autant, malgré l'ouverture de son code source, le partage d'*OXO* se heurte, à l'époque, à une réalité matérielle qui limitera considérablement la diffusion de ce jeu vidéo : en 1952, il n'existe qu'un seul ordinateur de type EDSAC à travers le monde, celui de l'université de Cambridge où Douglas effectue ses études...

Les laboratoires universitaires ont permis la naissance de plusieurs autres jeux sur ordinateur, tels que *Tennis For Two* (1958), un jeu de tennis pour deux joueurs, ou *HUTSPIEL* (1955), un jeu de stratégie au tour par tour destiné à la formation des généraux de l'armée américaine. Mais aussi innovants et intéressants qu'ils soient, ces jeux vidéo n'auront finalement qu'une influence très modeste dans l'histoire du jeu vidéo, du fait de leur diffusion confidentielle. Il faut alors attendre l'année 1962, pour voir apparaître un jeu vidéo « de laboratoire » qui a eu une influence notable : *Spacewar!*¹. Au début des années 1960, le Massachusetts Institute of Technology (MIT), célèbre université américaine spécialisée dans la recherche technologique, héberge parmi ses nombreux étudiants une poignée de génies animés par un esprit de création et d'innovation technologique : les *hackers*. Si aujourd'hui le terme hacker est souvent associé aux « pirates informatiques », dans les années 1960 il ne possède encore qu'une acceptation noble : il désigne des bidouilleurs experts en électronique et en informatique². Au MIT, le point de rassemblement des hackers

1. Damien DJAOUTI, « Les pionniers du jeu vidéo : Spacewar! », dans : *Pix'n Love* 12 (2010).

2. Steven LEVY, *Hackers. Heroes of the Computer Revolution*, New York : Dell Publishing, 1994.

est un club d'étudiants dédié au modélisme ferroviaire, le Tech Model Railroad Club (TMRC). Dans une grande pièce, ses membres passent des heures à créer les tracés ferroviaires les plus complexes et originaux qui soient. Si cette tâche ne nécessite *a priori* que des compétences en électronique, l'informatique deviendra rapidement un autre terrain d'expérimentation du club. Bidouillant tout d'abord un vieil *IBM 709* laissé à l'abandon dans les couloirs du MIT, les membres du TMRC s'épanouissent réellement à partir de 1959, année d'acquisition d'un ordinateur *TX-0* par le MIT. Équipé d'un écran CRT avec crayon optique, l'appareil est rattaché à l'Artificial Intelligence Group, un groupe de chercheurs en intelligence artificielle dirigé par les professeurs John McCarthy et Marvin Minsky. Leurs étudiants s'accaparent immédiatement la quasi-totalité des créneaux vacants sur le planning d'utilisation de la machine, soit la plage horaire allant de 23 h du soir à 7 h du matin. Une bonne partie d'entre eux étant des membres ou des sympathisants du Tech Model Railroad Club, le *TX-0* devient rapidement leur terrain d'expérimentation favori. Poussés par une recherche de la performance technique, ils y créent divers programmes, dont des jeux tels que *Mouse in a Maze* (Ross & Ward, 1961), *Tic-Tac-Toe* (Daly, 1961) ou encore *Qubic* (Daly, 1961). Fidèles à une philosophie de partage et de diffusion de la connaissance, tous les programmes des hackers sont « ouverts ». Leur code source est librement accessible, ce qui permet à chaque hacker d'améliorer les programmes et jeux écrits par d'autres, avant de partager à leur tour la version améliorée du logiciel. Concrètement, cela signifie que les cartes perforées sur lesquelles sont enregistrées les codes sources de ces différents programmes sont stockées dans un placard qui n'est volontairement pas fermé à clé, afin que tout le monde puisse y avoir accès¹. S'il est trop tôt pour parler d'*open source*, voire de logiciel libre², nous reconnaissons clairement ici une volonté d'ouverture et de partage de codes sources qui se retrouve aujourd'hui dans la philosophie du Libre. Pourtant, malgré cette ouverture, l'influence de ces quelques jeux créés par les hackers du TMRC resta limitée aux seuls membres du club.

1. Martin GRAETZ, « The Origin of Spacewar ! », dans : *Creative Computing 7.8* (1981), p. 56–67.

2. La formalisation des notions de Libre et d'*open source* n'ayant eu lieu qu'à partir des années 1980.

1.2 Spacewar !

À l'automne 1961, le *TX-0* se voit doté d'un compagnon de jeu, le *PDP-1* du constructeur DEC. Encore plus puissant que le *TX-0*, ce nouvel ordinateur est également mieux équipé. Plus petit, il est plus facile d'accès, sa mise en route pouvant être effectuée par une personne seule. Le *PDP-1* propose en standard un écran CRT, une imprimante, un crayon optique et surtout une mémoire impressionnante de 4K. Seul problème, comme le *TX-0* à son arrivée, il n'existe aucun programme pour cette machine. Ce terrain d'expérimentation vierge ne laissa pas insensible les hackers du TMRC, et les conduisit même à inventer un jeu vidéo pionnier qui, cette fois, a marqué l'Histoire : *Spacewar !*

Réalisé entre 1961 et 1962 pour le *PDP-1*, *Spacewar !* est un jeu de combat spatial pour deux joueurs. Le concept du jeu est imaginé en 1961 par trois étudiants du MIT qui habitent en collocation et fréquentent tous le TMRC : Stephen R. Russell, J. Martin Graetz et Wayne Wiitanen. Ce dernier doit malheureusement accomplir son service militaire avant que la réalisation du jeu ne commence. Steve Russel devient alors le programmeur principal, assisté par Martin Graetz. Après 200 heures de travail, les deux compères obtiennent une première version du jeu au mois de février 1962¹. Deux vaisseaux, un stock d'essence limité et une réserve de missiles, il s'agit d'une version épurée mais efficace du concept originel. Chaque joueur se déplace librement dans l'espace et doit essayer de détruire l'autre vaisseau avec ses missiles. En somme, il s'agit là du premier *shoot'em up*² de l'histoire³.

À partir de ce moment commence un processus de design itératif⁴, où *Spacewar !* évolue par petites touches : le jeu est testé par le groupe de hackers, on note les idées d'amélioration, on les implémente, puis on re-teste. Le premier problème ainsi identifié sur cette version du jeu est lié au fait que la partie se déroule sur un fond noir, ce qui rend difficile

1. *ibid.*

2. Jeu vidéo dans lequel on incarne un vaisseau spatial qui doit en détruire de nombreux autres à coup de lasers.

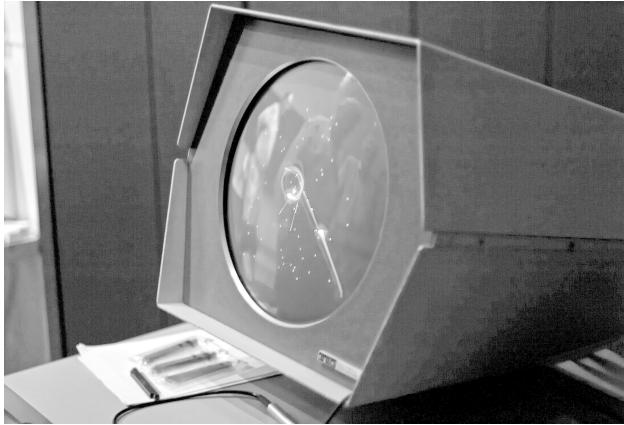
3. Steven L. KENT, *The Ultimate History of Video Games : From Pong to Pokemon—The Story Behind the Craze That Touched Our Lives and Changed the World*, Three Rivers Press, 2001.

4. Processus de création découpé en étapes successives visant à créer un objet de plus en plus complet.

l'appréciation de la vitesse des vaisseaux. Russell rajoute donc un fond étoilé aléatoire. Mais les hackers sont aussi des gens perfectionnistes. À ce titre, Peter Samson, un autre membre du TMRC, n'aime pas ce champ d'étoiles généré aléatoirement. Il se met alors en tête d'ajouter au programme la représentation des étoiles véritablement visibles dans le ciel, en respectant leur position et leur luminosité. Un rien moqueur, le groupe de hackers nomme cette fonctionnalité « Expensive Planetarium »¹, pour rappeler qu'ils utilisent un ordinateur coûtant 120 000 \$ afin de dessiner une simple carte du ciel. Cette anecdote illustre d'ailleurs à merveille la dimension « partage et amélioration participative », aujourd'hui pilier de la philosophie du Libre, qui se retrouve dès les années 1960 dans *Spacewar!*.

Cette philosophie du Libre changea ainsi la nature du jeu pour le faire évoluer vers la forme que nous connaissons aujourd'hui. En l'état, la première version manque quelque peu de finesse : le jeu consiste tout simplement à se déplacer le plus vite possible tout en tirant un maximum de missiles afin d'avoir une chance de toucher l'adversaire. Dan Edwards, également membre du club de hackers, rajoute ensuite la gestion de la gravité pour rendre le jeu plus stratégique. Ce faisant, il introduit un élément essentiel : l'étoile qui se trouve au centre de l'écran et qui attire inexorablement les vaisseaux des joueurs. Ce nouvel ajout chamboule complètement l'expérience de jeu, les joueurs devant maintenant faire attention à leurs déplacements, et penser à leur survie autant qu'à détruire l'adversaire. En d'autres termes, le jeu devient alors un excellent compromis entre action et stratégie, comme en témoigne son succès au sein de groupe. Martin Graetz se charge de rajouter une dernière fonctionnalité au jeu : le déplacement en hyper-espace. En appuyant sur un bouton dédié, le vaisseau disparaît de la scène de jeu et réapparaît à un endroit aléatoire. Très pratique pour éviter un missile trop proche, cette option n'est pas sans risque car elle peut vous faire réapparaître au beau milieu de l'étoile centrale, synonyme de mort immédiate. Ainsi, au mois d'avril 1962, *Spacewar!* est terminé. Mais si le jeu en lui-même est aussi révolutionnaire qu'amusant, son maniement à partir de boutons à presser sur le *PDP-1* n'est pas des plus pratiques. Qu'à cela ne tienne, Alan Kotok et Bob Saunders, membres

1. « Planétarium hors de prix ».



Spacewar! tournant sur un PDP-1. (Par Joi Ito, sous licence CC-BY 2.0)

éminents du TMRC, utilisent des pièces de modélisme ferroviaire pour bricoler des boîtiers de commandes adéquats !

Ce jeu est un succès incontesté au sein du club, les hackers passant dorénavant de nombreuses heures à jouer à *Spacewar!* quand il ne programment pas. Ce fut également un succès lors des journées portes ouvertes du MIT, en mai 1962, événement pour lequel un système de score fut rajouté au jeu afin de limiter la durée de chaque partie¹. Si les acteurs principaux de ce monument de l'histoire du jeu vidéo finissent par quitter le MIT, la nature ouverte du jeu lui assure une vie propre. Il se répand au sein des diverses universités américaines, de nouvelles variantes (ciel qui tourne, missiles sensibles à la gravité de l'étoile, boucliers. . .) étant créées à chacun de ses voyages. Mais ce jeu connaît également une application « utilitaire » on ne peut plus inattendue. DEC, le constructeur du *PDP-1*, était en étroite relation avec les étudiants du MIT, et n'hésitait pas à leur fournir des programmes². En échange, le constructeur récupérait le travail de ces hackers de génie et l'exploitait comme bon lui semblait. Ainsi, *Spacewar!* fut logiquement mis en avant par le service commer-

1. GRAETZ, *op. cit.*

2. DEC a, par exemple, fourni des routines de fonctions « sinus / cosinus » que Russel utilisera pour *Spacewar!*.

cial de DEC pour vanter les capacités de sa machine¹. Mais surtout, il fut utilisé comme programme de test pour vérifier son bon fonctionnement. Contrairement à la RAM de nos ordinateurs actuels, la mémoire centrale du *PDP-1* n'était pas volatile, ce qui signifie qu'un programme qui y est chargé reste présent après l'extinction de l'appareil. Au sortir de la chaîne de fabrication du *PDP-1*, les ingénieurs chargeaient donc *Spacewar!* dans la mémoire de l'ordinateur. Après livraison de la machine au client, l'installateur l'allumait et, si le jeu était encore en mémoire et parfaitement fonctionnel, cela prouvait que la machine n'avait pas subi de problème matériel lié à son transport ou à son installation².

1.3 De la libre diffusion à la logique propriétaire : la naissance d'une industrie

De par sa large diffusion au sein des universités américaines, nombreuses sont les personnes à avoir pu jouer à *Spacewar!*. Parmi ces joueurs, certains furent inspirés et se lancèrent dans la réalisation de leur propre jeu vidéo. Mais ceux-là étaient animés par un esprit plus proche de la recherche de profit qui motive les entrepreneurs que celui de partage propre aux hackers³.

Le premier jeu influencé par *Spacewar!* est, à ce jour, le plus ancien jeu vidéo de type « borne d'arcade » connu. Il s'agit tout simplement d'une version du code originel de *Spacewar!* qui a été reprogrammée sur un ordinateur *PDP-11*, une machine bien plus compacte et économique que le *PDP-1*⁴. Les auteurs du jeu, Bill Pitts et Hugh Tuck, le bricolent alors pour qu'il puisse recevoir des pièces de monnaie, et modifient le jeu en conséquence. Après trois mois et demi de dur labeur, *Galaxy Game* (1971) est né. En septembre 1971, les deux bidouilleurs installent la machine dans un bar de l'université de Stanford. Au tarif attractif de 10 cents pour une partie ou 25 cents pour trois parties, la machine rencontre un grand

1. Heather CHAPLIN et Aaron RUBY, *Smartbomb : The Quest for Art, Entertainment, and Big Bucks in the Videogame Revolution*, New York : Algonquin Books, 2005.

2. LEVY, *op. cit.*

3. Matt BARTON et Bill LOGUIDICE, « The History of Spacewar! : The Best Waste of Time in the History of the Universe », dans : *Gamasutra* (2009), URL : <http://www.gamasutra.com/view/feature/4047/>.

4. « Seulement » 20 000 \$ pour un PDP-11 au lieu des quelques 120 000 \$ d'un PDP-1.



Computer Space, un jeu vidéo commercial et propriétaire largement inspiré par *Spacewar!* (Par Rob Boudon, sous licence CC-BY 2.0.)

succès auprès des étudiants. Au plus haut de son succès, la durée d'attente pour y jouer dépasse allègrement l'heure tellement les clients sont nombreux. Une seconde version de la borne voit donc le jour en 1972. Dotée de plusieurs écrans, elle permet d'accueillir jusqu'à huit joueurs. Le jeu restera un énorme succès jusqu'à son retrait définitif en 1979, pour cause de dommages matériels irréparables. La borne fut alors démontée et mise au rebut, avant d'être miraculeusement restaurée en 1997 pour le Computer History Museum de l'université de Stanford¹.

Mais la plus grande influence de *Spacewar!* sur l'histoire du jeu vidéo découle directement de l'énorme impression qu'a eu ce titre sur un étudiant du nom de Nolan Bushnell. Ce dernier est tellement marqué par le jeu qu'il copie également son concept pour créer la première borne d'arcade largement commercialisée, *Computer Space* (1971). En effet *Galaxy Game* tourne sur un matériel proche de celui de *Spacewar!*, dont les coûts de fabrication l'ont empêché d'être produit à grande échelle. Pour *Com-*

1. Bill PITTS, « The Galaxy Game », dans : *Stanford Computer History Exhibits* (1997), URL : <http://infolab.stanford.edu/pub/voy/museum/galaxy.html>.

puter *Space*, Bushnell recrée donc le jeu sur du matériel moins onéreux, mais se voit contraint de le simplifier au passage. Il retire par exemple l'étoile centrale et son champ de gravité, mais rajoute un mode mono-joueur. Le principe du jeu consiste maintenant à diriger une fusée pour tirer sur des soucoupes volantes sans se faire toucher. Une seconde version de la borne propose un nouveau mode de jeu dans lequel les soucoupes volantes peuvent être dirigées par un deuxième joueur. Mais, malgré les espoirs du jeune entrepreneur, la borne commercialisée en novembre 1971 est un échec commercial. Pourtant, Bushnell croit en l'idée de commercialiser des jeux vidéo, sous forme de logiciels et machines propriétaires, au grand public. Il persévère et fonde, avec son associé Ted Dabney, la société Atari en juin 1972. Inspiré par une présentation de la *Magnavox Odyssey*¹, Bushnell invente un nouveau concept de jeu, dont il confiera la réalisation au premier employé d'Atari, Alan Alcorn. Ce nouveau jeu d'arcade, baptisé *Pong* (1972), connaît alors un succès retentissant avec près de 350 000 unités vendues. Créée en 1972, la société Atari affiche déjà 11 millions de dollars de chiffre d'affaires en 1973, et plus de 36 millions en 1975². À ce titre, bien qu'il ne soit pas le premier jeu vidéo, ni le premier jeu d'arcade, et que son principe fut visiblement inspiré du jeu *Ping-Pong* de la *Magnavox Odyssey*, *Pong* est généralement considéré comme le point de départ de l'industrie du jeu vidéo³. En effet, il est le premier à avoir connu un véritable succès commercial de masse. Tout cela serait-il arrivé si Bushnell n'avait pas autant joué à *Spacewar!* alors qu'il était étudiant ? On peut légitimement en douter. Mais l'entrepreneur aurait-il été en mesure d'y jouer si ce jeu n'avait été pas été créé selon une

1. La *Magnavox Odyssey* est la première console de jeu vidéo de salon commercialisée auprès du grand public, tout du moins aux États-Unis. Inventée par Ralph Baer, il s'agit d'un appareil qui peut se brancher sur un téléviseur afin de jouer à différents jeux vidéo simples, grâce à deux manettes reliées à l'appareil. Véritable innovation technologique à l'époque, elle propose parmi les jeux un certain *Ping-Pong*. Ce dernier permet à deux joueurs de s'affronter dans un jeu où il faut renvoyer une balle en contrôlant des raquettes rectangulaires. La console fut présentée au public avant sa commercialisation. En mai 1972, Bushnell a assisté à la présentation de l'*Odyssey*, soit un mois avant qu'il ne fonde la société Atari et ne définit le concept du jeu *Pong*. Si *Pong* et *Ping-Pong* ne se sont pas en tout point identiques (par exemple *Ping-Pong* ne possède pas de score et n'utilise pas un déplacement limité à l'axe vertical comme *Pong*), force est de constater que les deux jeux se ressemblent beaucoup... *Magnavox* portera d'ailleurs plainte contre Atari à ce sujet dès 1974.

2. Michael LEARMONTH, « No Pain, No Game », dans : *Metroactive.com* (1999), URL : <http://www.metroactive.com/papers/metro/09.16.99/cover/bushnell2-9937.html>.

3. Tristan DONOVAN, *Replay : The History of Video Games*, Yellow Ant, 2010.

philosophie permettant sa libre diffusion ? Après tout, Bushnell était étudiant à l'université de l'Utah, située à plus de 4 000 kilomètres du MIT. Comme nous l'avons vu, le fait que *Spacewar!* soit un logiciel ouvert, préfigurant ainsi les logiciels libres actuels, a grandement participé à sa large diffusion, que ce soit entre étudiants ou par le biais du constructeur *DEC*. De plus, Bushnell aurait-il été en mesure de programmer son jeu *Computer Space* s'il n'avait pas pu préalablement étudier le code source de *Spacewar!* ? En d'autres termes, aurait-il tout simplement pu créer lui-même un jeu vidéo si *Spacewar!* avait été un logiciel propriétaire ?

En guise de réponse, nous pouvons nous référer une célèbre citation de Bushnell, qui laisse deviner que l'ouverture des codes sources et la libre distribution des premiers jeux vidéo l'ont clairement aidé à donner naissance à une industrie aujourd'hui florissante : « Je n'ai pas inventé les jeux vidéo. Ils ont été inventés sur des ordinateurs à 7 millions de dollars. Moi, je n'ai fait que les commercialiser. »¹

2. Du shareware au modding

2.1 Le *shareware* : libérer les canaux de distribution de jeux vidéo

Attirées par l'incroyable succès de Bushnell et sa société Atari, de nombreuses autres entreprises s'intéressent au jeu vidéo, et proposent des jeux et des machines de plus en plus sophistiqués. Au-delà de l'innovation technologique et artistique, le passage des premiers jeux vidéo commercialisés en 1972 au secteur industriel générant aujourd'hui un chiffre d'affaires estimé à plusieurs milliards de dollars² s'est également accompagné de la disparition progressive de toute trace de la philosophie du Libre qui animait les créateurs de *Spacewar!*. La commercialisation des logiciels de jeu, qu'il s'agisse de cartouches de jeu pour ordinateur ou console de salon, ou bien de bornes d'arcade, a été synonyme de la fermeture de l'accès à leur code source. De même, la liberté d'exécution s'est trouvée limitée par le fait de ne pouvoir utiliser certains logiciels de jeu que sur certaines machines. Si Atari a tracé les principales lignes de ce modèle économique reposant sur une logique propriétaire, elle n'a pas été la

1. Nous traduisons. LEARMONTH, *op. cit.*

2. Voir http://www.afjv.com/news/839_sales-video-game-content.htm.

seule¹. En effet, lors de la sortie de sa première console de jeu vidéo de salon, la *VCS 2600* (1977), Atari pensait logiquement qu'elle serait la seule à développer des jeux pour sa machine. Quelle ne fut pas alors sa surprise de constater que d'autres sociétés, attirées par le succès commercial de la console, ont – elles aussi – commencé à fabriquer des cartouches de jeu, et ce, sans autorisation de la part d'Atari. La société qui a ouvert la brèche s'appelle Activision, un studio indépendant de création de jeu vidéo fondé par quatre ex-employés d'Atari. Malgré les protestations d'Atari, la pratique fut déclarée légale car Activision n'enfreignait aucun brevet propre à Atari. Cette « erreur » fut réparée par Nintendo lors de la génération suivante de consoles. En effet, les cartouches de jeux pour la console *Nintendo NES* (1983) embarquent une puce de protection, la *10NES*². Cette puce au code breveté garantit que seul Nintendo peut légalement produire des cartouches de jeu pour sa console. Ce faisant, l'industrie du jeu vidéo a réussi à non seulement limiter la liberté des utilisateurs de logiciels, mais aussi celle des créateurs. Cette politique est toujours d'actualité aujourd'hui et les créateurs de jeu vidéo doivent recevoir une autorisation (payante) de la part des constructeurs de consoles de jeu afin de pouvoir développer pour elle. Un modèle similaire se poursuit dans certains circuits de distributions dématérialisés. Par exemple, le *Xbox Live Arcade* dont les portes d'entrée sont contrôlées par Microsoft, ou encore Steam dont seul le propriétaire, Valve, peut autoriser un jeu à y figurer. Enfin, si le développement de jeux vidéo sur ordinateur a toujours été accessible à toute personne possédant des connaissances techniques en la matière, le fait d'éditer, de fabriquer et de commercialiser un jeu dans un réseau de magasins physiques représente, depuis la fin des années 1980, un coût de plus en plus important³. Cela limite donc considérablement la possibilité pour tout créateur de jeu vidéo indépendant d'y distribuer ses réalisations.

D'autres circuits se mettent alors en place pour permettre une diffusion plus libre de jeux vidéo. Optant pour la distribution de logiciels sous forme dématérialisée, ces circuits parallèles s'appuient tout d'abord sur

1. KENT, *op. cit.*

2. Voir <http://en.wikipedia.org/wiki/10NES> pour plus de détails.

3. D'après des données américaines datant de 2010, la moitié du prix d'un jeu vendu en magasin est consacrée à sa distribution, l'autre moitié couvrant les coûts de développement et de marketing : <http://latimesblogs.latimes.com/entertainmentnewsbuzz/2010/02/anatomy-of-a-60-dollar-video-game.html>.

des serveurs BBS¹, puis sur Internet. Deux principaux types de logiciels circulent sur ces réseaux. D'un côté les *freeware*, logiciels intégralement gratuits, et de l'autre les *shareware*, dont une version incomplète est distribuée gratuitement dans l'espoir que vous en commandiez la version complète et payante directement au créateur. Si tous les types d'applications se retrouvent sur ces réseaux, les jeux vidéo y sont largement représentés. Par rapport aux autres circuits de distribution, les réseaux BBS et Internet sont relativement « libres », car tout le monde ou presque peut y distribuer ses créations. Cette liberté permet donc enfin à des jeux vidéo optant pour une philosophie différente du modèle propriétaire d'exister. Certains *freeware* sont alors parfois distribués avec leur code source sur ces réseaux, même si cela n'est pas la norme².

Dans les années 1980, au-delà du développement des réseaux informatiques, nous assistons également à la formalisation des pratiques d'ouverture de code source et de libre distribution de logiciels sous forme de licences juridiques. Nous pensons à la célèbre licence GNU/GPL, dont la première version officielle est publiée en 1989, ou aux premières versions de la BSD licence qui remontent à 1988. Si le courant du logiciel libre associe l'ouverture de code source à une éthique prônant le partage et la libre distribution, le courant de l'*open source*, apparu quelques années plus tard, s'arrête quant à lui au seul fait que le code source d'un logiciel soit ouvert³. Ainsi, qu'ils soient présentés comme logiciels libres, comme *open source*, ou qu'il s'agisse tout simplement d'un programme au code source ouvert publié avant toute formalisation d'une philosophie du Libre, de nombreux jeux vidéo ont pu être diffusés sur les réseaux BBS et Internet depuis la fin des années 1970 jusqu'à nos jours. Mais ces modes de

1. Un Bulletin Board System est un ordinateur auquel d'autres ordinateurs peuvent se connecter par le biais d'un modem. Une fois connecté à un BBS par le réseau téléphonique, un utilisateur peut discuter avec d'autres membres, envoyer ou récupérer des fichiers, jouer à des jeux en réseau... Il s'agit en quelque sorte de l'ancêtre d'Internet et de ses nombreux services. Les BBS rencontreront un certain succès auprès des passionnés d'informatique, à une époque où le « réseau des réseaux » n'était pas encore accessible au grand public (soit de la fin des années 1970 au milieu des années 1990).

2. Par exemple, sur les 777 jeux vidéo indépendants référencés sur <http://db.tigsources.com/>, seulement 8 sont distribués avec leur code source. Cela représente uniquement 1 % des titres de cette base de données recensant des jeux vidéo distribués exclusivement par voie dématérialisée.

3. Pour plus de détails à ce sujet, voir par exemple http://fr.wikipedia.org/wiki/Logiciel_libre.

distribution électronique n'ont pas uniquement contribué au retour des jeux vidéo ouverts sous la forme de logiciels libres ou *open source*. Ils ont également largement contribué au développement du piratage de logiciels, phénomène qui n'a bien évidemment pas épargné les jeux vidéo. Cette thématique sort du cadre de ce chapitre, mais notons néanmoins que, dès le début des années 1980, les réseaux BBS puis Internet ont permis à des groupes de *crackers* de distribuer gratuitement des jeux vidéo commerciaux débarrassés de tout dispositif technique de protection contre la copie. Si les motivations des *crackers* pour s'investir dans une telle activité, de nature illégale dans la plupart des pays, sont nombreuses et variées, le fait de « libérer des logiciels propriétaires » en fait généralement partie¹.

En résumé, l'apparition de nouveaux modes de distribution a tout d'abord contribué au retour des jeux vidéo ouverts, qui rejoignent ensuite les rangs du logiciel libre ou des logiciels *open source* avec la formalisation de ces philosophies. Mais ces réseaux ont également permis l'apparition du piratage des nombreux jeux vidéo reposant toujours sur le modèle du logiciel propriétaire. En effet, malgré l'apparition des BBS et d'Internet, le logiciel propriétaire reste le modèle de référence depuis les prémices commerciales de son industrie. Cependant, au-delà de ces deux impacts majeurs, les réseaux BBS et Internet aident également à ré-insuffler un peu de la philosophie de partage et de création collaborative chère au Libre au sein des jeux vidéo, en particulier du côté des *shareware*² avec l'apparition de la pratique du *modding*.

2.2 Le modding

Comme son nom l'indique, le *modding* désigne le fait de modifier un jeu existant, pour ensuite redistribuer le fruit de ce travail, baptisé un *mod*. Mais contrairement à un logiciel libre dont on modifierait directement le

1. Paul CRAIG, *Software Piracy Exposed*, Waltham : Syngress, 2005.

2. Le *shareware*, ou « partagiciel », désigne un mode de distribution de logiciels qui consiste à diffuser gratuitement une version fonctionnelle mais limitée de son programme pour donner envie à l'utilisateur d'en acheter une version complète. Dans le cas du jeu vidéo, il s'agit de titres, par exemple *Doom* (id Software, 1993), dont les premiers niveaux sont diffusés gratuitement, le joueur devant ensuite acheter les niveaux suivants directement aux créateurs du jeu.

code source pour ensuite redistribuer librement la version modifiée, un *mod* n'est pas autonome. Il ne contient qu'un ensemble de modifications effectuées sur un jeu donné. La possession du jeu originel est donc requise pour jouer à un *mod*. Qu'est-ce qui peut donc justifier un principe de fonctionnement aussi complexe ?

Tout simplement le fait que, contrairement aux logiciels libres, les jeux vidéo pour lesquels sont créés des *mods* sont généralement des logiciels propriétaires et commerciaux. Concrètement, les *mods* créés ne peuvent être commercialisés et sont parfois de nature *open source*, mais pour pouvoir jouer à un *mod* donné il faut auparavant acheter un jeu vidéo commercial et propriétaire. Prenons pour exemple le jeu vidéo de type *mod* le plus connu, à savoir *Counter-Strike* (Minh Le & Jess Cliffe, 1999). Ce jeu de tir en vue subjective a été le jeu vidéo multijoueur le plus pratiqué sur Internet pendant la première moitié des années 2000¹. Distribué gratuitement par le biais d'Internet, son installation nécessite d'avoir au préalable acheté et installé le jeu *Half-Life* (Valve, 1998), le jeu d'origine à partir duquel *Counter-Strike* a été créé. Si *Counter-Strike* a été réalisé par des amateurs éclairés sur leur temps libre, *Half-life* est, par contre, la création d'un studio de développement professionnel. Si les créateurs de *Half-life* autorisent ainsi tout amateur à créer et redistribuer librement des variantes de leur jeu, c'est parce que l'utilisation de ces variantes s'accompagne obligatoirement de l'achat du jeu originel. Ceci explique pourquoi les créateurs de *Half-life* ont, sur le cédérom de leur jeu, inclut des versions simplifiées des outils qu'ils ont utilisés lors de son développement. En distribuant ainsi leurs outils de travail, ils encouragent explicitement les amateurs à créer des variantes de leur propre jeu, qui seront ensuite distribuées sous forme de *mods*. Au final, si l'industrie du logiciel utilitaire arrive à faire cohabiter Libre et activité commerciale à travers la vente de services (formation, support), il semble que l'industrie du jeu vidéo a, de son côté, trouvé une autre voie à travers la pratique du *modding*. Philosophiquement, nous retrouvons dans ce modèle les idées de partage et d'amélioration de logiciels propre au logiciel libre, idées qui semblent avoir quitté l'industrie du jeu vidéo après *Spacemar!*. Alors, comment un développeur de jeux vidéo commerciaux et propriétaires aurait-il pu avoir

1. D'après <http://store.steampowered.com/stats/>.

un jour l'idée « folle » de distribuer librement ses outils de travail afin d'autoriser n'importe qui à bidouiller ses propres créations ?

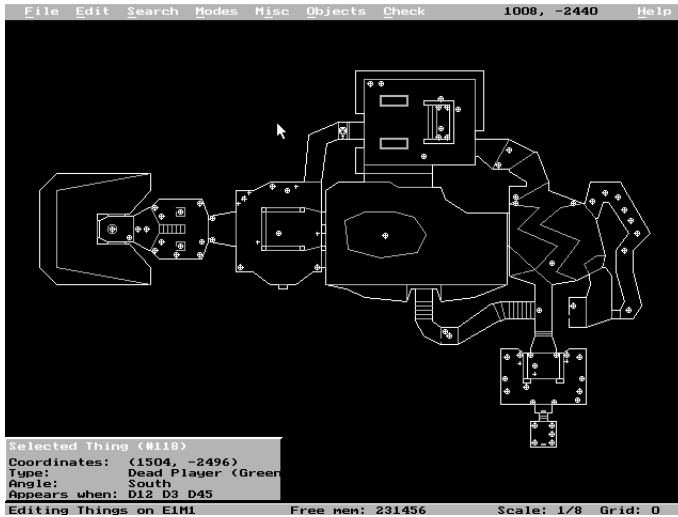
2.3 Le *shareware* comme berceau du *modding*

La réponse à cette question se trouve parfaitement illustrée par l'histoire d'un jeu vidéo en particulier, distribué sous forme de *shareware* : *Doom* (id Software, 1993). Il s'agit d'un jeu de tir en vue subjective, qui a marqué l'histoire du jeu vidéo en définissant les codes de ce genre vidéoludique aujourd'hui très populaire. La société à l'origine de *Doom*, id Software, est un petit studio indépendant dont les créations sont exclusivement distribuées sur serveur BBS selon les modalités du « *shareware* ». Tout commence lors de la publication de leur jeu *Wolfenstein 3D* (id Software, 1992). Les créateurs d'id Software sont très surpris de constater que de nombreux joueurs créent de nouveaux niveaux et des versions modifiées de ce titre, qu'ils s'échangent ensuite librement sur BBS¹. Mais ils ne cherchent pas à réprimander ces joueurs qui ont, en toute illégalité, « *hacké* » leur création afin de pouvoir la modifier. Au contraire, id Software encourage (et, quelque part, encadre) cette pratique pour le prochain titre alors en cours de réalisation : *Doom*.

Ainsi, le programmeur du jeu, John Carmack, invente le format de fichier *WAD*², afin de séparer le « contenu » du jeu de son « moteur ». Cela permet dorénavant aux joueurs de créer facilement du nouveau contenu pour *Doom* sans avoir à hacker le jeu. Il s'agit là d'un des premiers exemples de légitimation de la pratique visant à la modification de jeux vidéo commerciaux par des amateurs, par la suite appelée *modding*. Mais cette légitimation ne s'est pas faite sans contrepartie. En effet, id Software demanda explicitement à ce que le contenu créé par les joueurs ne puisse fonctionner que sur la version complète et payante du jeu, et non sur la version de démonstration gratuite (principe du *shareware*). Le hasard fit que la sortie de ce titre, qui rencontrait déjà un certain succès critique et commercial, s'accompagna du développement d'Internet pour le grand public. Ainsi, les amateurs qui créaient de nouveaux niveaux, monstres,

1. David KUSHNER, *Masters of Doom : How Two Guys Created an Empire and Transformed Pop Culture*, Random House Trade Paperbacks, 2004.

2. Acronyme de *Where's All the Data ?*



Doom Editing Utility, le premier éditeur de niveau pour *Doom* accessible à tous.

armes et autres variantes de *Doom* pouvaient se les échanger à travers des sites de fans tels que *Doomworld* (1993-2012) et *Gamers.org-DoomGate* (1994-2012). Pourtant, id Software ne livra pas ses outils de développement avec *Doom*, bien qu'elle encourageât les joueurs à modifier son titre. En effet, l'éditeur de niveaux utilisé par id Software pour créer *Doom* ne tourne que sur les ordinateurs du constructeur NeXT, et il est donc inaccessible à tous les joueurs sur PC (Ms-Dos). Les outils de développement n'étant pas accessibles aux joueurs pour de simples questions techniques, ceux-ci s'empressèrent de créer de nombreux outils par eux-mêmes. Par exemple, *Doom Editing Utility* (Brendon Wyber, 1994), diffusé un mois seulement après la sortie du jeu, permet d'en modifier les niveaux, tandis que *DeHacked* (Greg Lewis, 1994), sert à en altérer les règles.

Il faut ensuite attendre la sortie de *Duke Nukem 3D* (3D Realms, 1996) pour qu'un développeur livre directement tous ses outils de développement avec le jeu. D'après Scott Miller, un des créateurs de *Duke Nukem 3D*, cette décision était courageuse à une époque où l'industrie du jeu vidéo pensait qu'il ne fallait absolument pas diffuser ses outils et tech-

nologies¹. L'histoire montre qu'à l'inverse, cette pratique augmente le potentiel commercial du jeu en prolongeant sa durée de vie. Ainsi, plus de quinze ans après sa sortie, *Doom* est toujours joué et utilisé comme support de création. De nouveaux éditeurs de niveaux continuent même à voir le jour pour ce titre, à l'image *Doom Builder 2* (Pascal vd Heiden, 2009). À ce jour, *Counter-Strike* semble être le premier *mod* dont le succès populaire dépasse largement celui de son jeu d'origine. Ce succès permet au grand public de découvrir la pratique du *modding*, qui restait jusqu'alors cantonnée à une certaine frange de joueurs passionnés². Aujourd'hui, la création de *mods* représente une valeur marchande indirecte pour les éditeurs de jeux vidéo³, à tel point que certains professionnels de l'industrie intègrent dorénavant la création amateur de *mods* dans leur stratégie commerciale⁴.

3. La création amateur de jeu vidéo : de l'*open source* au Jeu 2.0

Parallèlement à l'histoire du jeu vidéo commercial, et de l'industrie qu'il fait vivre, la création de jeu vidéo est également pratiquée en amateur depuis de nombreuses années. Comme nous allons le voir, l'influence du Libre sur la création vidéoludique amateur est assez différente de celle que nous venons d'étudier du côté industriel.

3.1 L'échange de code source

L'histoire de la création de jeu vidéo en amateur est intimement liée à celle de l'informatique, et en particulier au développement de la micro-informatique personnelle⁵. Un des premiers micro-ordinateurs à avoir

1. D'après une interview personnelle de Scott Miller réalisée par mail en mai 2010.

2. Tero LAUKKANEN, « Modding Scenes - Introduction to user-created content in computer gaming », dans : *Hypermedia Laboratory Net Series* 9 (2005), URL : <http://tampub.uta.fi/handle/10024/65431>.

3. Hector POSTIGO, « Of Mods and Modders Chasing Down the Value of Fan-Based Digital Game Modifications », dans : *Games and Culture* 2.4 (2007), p. 300-313, URL : <http://gac.sagepub.com/content/2/4/300>.

4. David B. NIEBORG et Shenja van der GRAAF, « The mod industries ? The industrial logic of non-market game production », dans : *European Journal of Cultural Studies* 11.2 (2008), p. 177-195, URL : <http://ecs.sagepub.com/content/11/2/177>.

5. CHAPLIN et RUBY, *op. cit.*

rencontré un succès commercial auprès du grand public est l'*Altair 8800* (MITS, 1975). Sorti en 1975, cet ordinateur avait la particularité d'être vendu en kit. Il a donc rapproché de nombreux passionnés d'informatique, qui se sont rapidement regroupés au sein de « clubs » afin d'échanger des conseils d'assemblage. Mais ces clubs étaient aussi un excellent moyen d'échanger des programmes pour cette machine. L'un des rares programmes disponibles lors de sa sortie était l'*Altair Basic* (Micro-Soft, 1975), première création commerciale d'une jeune société qui sera plus tard connue sous le nom de Microsoft. Les détenteurs d'un *Altair 8800* pouvaient donc écrire leurs propres programmes en langage BASIC. Si seuls certains de ces clubs deviendront célèbres, à l'image du Homebrew Computer Club fréquenté par Steve Jobs et Steve Wozniak, tous avaient la particularité de permettre la circulation de nombreux petits jeux vidéo écrits en BASIC¹. Réalisés par des amateurs, ils s'échangeaient directement sous forme de listing BASIC. Cette pratique informelle a rapidement été accompagnée par la publication d'ouvrages regroupant des codes sources, tel que *Basic Computer Games*². Écrit par le journaliste et spécialiste de l'histoire du jeu vidéo David Ahl, cet ouvrage rassemble les codes sources de 101 jeux. Il a connu plusieurs éditions pour diverses machines. La première date de 1973 et est consacrée au BASIC tournant sur les ordinateurs de la famille *PDP* du constructeur DEC. La version de 1978, consacrée au BASIC de Microsoft, rencontra un énorme succès auprès du public puisque plus d'un million d'exemplaires furent vendus au total. L'ouvrage fut même traduit en français par Sybex en 1980. À noter que l'année 2010 vit une réédition de cet ouvrage pour le langage Smallbasic³. Au fur et à mesure de la sortie de nouveaux modèles de micro-ordinateurs de plus en plus performants, tels que l'*Apple II* (Apple, 1977), la création et l'échange de jeux amateurs sous forme de codes sources se développa considérablement. Certains magazines se spécialisèrent même dans cette activité, à l'image du français *Hebdogiciel* (Shift Éditions, 1983-1987).

1. Mark J. P. WOLF, éd., *The Video Game Explosion : A History from PONG to PlayStation and Beyond*, Westport : Greenwood, 2007.

2. David H. AHL, *BASIC Computer Games : Microcomputer Edition*, Microcomputer ed, New York : Workman Pub Co, 1978

3. <http://smallbasiccomputergames.com/>.

Cependant, à partir de la fin des années 1970, la plupart des micro-ordinateurs étaient équipés d'un lecteur de disquettes ou de cassettes. Cela permettait donc également aux amateurs d'échanger leurs créations vidéoludiques directement sous forme électronique. Le fait de ne plus être obligé de partager le code source pour diffuser leurs créations semble avoir détourné de nombreux amateurs de la voie du Libre. Si la création de jeux amateurs en *open source* restait bien évidemment populaire, elle n'était plus la seule voie possible. Mais la diffusion de jeux amateurs directement sous forme logicielle amène une nouvelle problématique. En effet, même avec le langage BASIC, la création de jeux vidéo reste une tâche relativement complexe, en particulier pour des débutants. Certains membres des communautés amateurs ont donc imaginé des outils destinés à simplifier la création de jeu vidéo : les « usines à jeux ».

3.2 Les usines à jeux

Devant la complexité que représente la réalisation d'un nouveau jeu vidéo pour des amateurs, certains programmeurs ont imaginé des outils logiciels permettant de faciliter leur création. Le plus ancien que nous ayons pu recenser à ce jour est *Eamon* (Brown, 1980). Explicitement destiné à une diffusion non commerciale, ce programme est un jeu de rôle et d'aventure en mode texte, dans la grande tradition de *Colossal Cave Adventure* (Crowther & Woods, 1976). Mais surtout, il est livré avec des utilitaires permettant, y compris pour un non-programmeur, de créer ses propres aventures qui sont ensuite redistribuables sous forme de jeux autonomes¹. Ce genre d'outil logiciel s'est considérablement développé par la suite, structurant la sphère des créateurs amateurs en « communautés » centrées sur un logiciel donné. Nous appelons de tels logiciels des « usines à jeux », que nous pouvons définir comme un logiciel « tout-en-un » permettant de créer un jeu vidéo autonome sans forcément partir d'une base existante. La notion « d'usine » se justifie par cette idée d'avoir un logiciel « tout-en-un », centralisant ainsi une chaîne de production complète de jeux vidéo. Bien que principalement utilisés par les amateurs, des outils similaires sont utilisés dans l'industrie à des fins de réduction de coûts de production par économie d'échelle.

1. Frank KUNZE et Greg GIOLA, « A Brief Overview of Eamon », dans : *Eamon Adventurer's Guild Online* (2008), URL : <http://www.eamonag.org/pages/eamonoverview.htm>.

Après *Eamon*, d'autres usines à jeux sont rapidement apparues pour proposer la création d'autres genres de jeux. Ces derniers s'appuient d'ailleurs sur une représentation graphique et sonore plus évoluée en utilisant des graphismes en 2D au lieu du texte. L'outil le plus emblématique est sans doute *Pinball Construction Set* (Budge, 1983). Comme son nom l'indique, ce logiciel pionnier permet de créer des jeux de flipper. Ses divers éditeurs permettent de composer une table de flipper en posant divers éléments (bumpers, trous...), puis d'en définir les règles de bases (nombre de balles...). Le jeu est ensuite exportable sous forme autonome, ce qui permet de le diffuser librement. Le succès critique et commercial rencontré par cet outil a inspiré la création de nombreux logiciels similaires, toujours destinés à la création d'un genre vidéoludique en particulier¹. Par exemple, *Adventure Construction Set* (Smith, 1984) permet de réaliser des jeux de rôle et d'aventure dans la veine des premiers opus de la série *Ultima* (Garriot, 1980-2009). De son côté, *Racing Destruction Set* (Koenig, 1985) permet de créer des jeux de course, alors que *War-game Construction Set* (Strategic Simulations Inc, 1986) se focalise sur les jeux de stratégie militaire au tour par tour. Si toutes ces usines à jeux sont spécialisées dans la création d'un genre en particulier, il en existe d'autres qui permettent de créer tous les genres vidéoludiques, voire d'en inventer de nouveaux. Le pionnier en la matière est sans conteste *Game-maker* (Kitchen, 1985), qui fut ensuite suivi par deux familles de logiciels aujourd'hui très populaires. D'un côté, les logiciels créés par la société Clickteam, qui rassemblent notamment *Klik & Play* (1994), *The Games Factory* (1998) ou encore *Multimedia Fusion 2* (2006). De l'autre, les différentes versions de *Game Maker* (Overmars, 1999-2012). Ces deux familles d'outils sont destinées à la création de jeux vidéo en 2D. Mais nous pouvons également citer la gamme des *3D Game Studio* (Conitec, 1993-2010), *BlitzMax* (Blitz Research, 2004) ou encore les différentes versions d'*Alice* (Pausch & al., 1995-2010) qui permettent la création de jeux vidéo en 3D. La place nous manque pour rentrer dans le détail historique des très nombreuses usines à jeux existantes. Nous renvoyons donc le lecteur intéressé par le sujet au site *Game Creation Tools Classification*²,

1. Matt BARTON et Bill LOGUIDICE, « The History of the Pinball Construction Set : Launching Millions of Creative Possibilities », dans : *Gamasutra* (2010), URL : <http://www.gamasutra.com/view/feature/3923>.

2. <http://creatools.gameclassification.com>.

une base de données collaborative en ligne qui recense plus de 450 usines à jeux publiées depuis 1980.

Cependant, qu'elles soient disponibles gratuitement ou diffusées commercialement, force est de constater que la plupart des usines à jeux sont des logiciels propriétaires. Cela peut paraître regrettable car, comme c'est le cas des jeux commerciaux permettant le *modding*, l'ouverture du code source d'une usine à jeux semble prolonger sa durée de vie. C'est par exemple le cas de *MegaZeux* (Janson, 1994), destiné à faciliter la création de jeux d'action-aventure utilisant des graphismes ASCII¹. Originellement diffusée sous forme de *shareware* en 1994, la carrière de cette usine à jeux fut prolongée par son passage dans le domaine public en 1999. Le développement de *MegaZeux* a alors été poursuivi par une communauté d'amateurs passionnés. Aujourd'hui, de nombreux jeux continuent à être créés avec *MegaZeux*, grâce une communauté d'utilisateurs restreinte mais toujours active plus de quinze ans après la sortie du logiciel. *Official Hamster Republic Role Playing Game Creation Engine* (Paige, 1997) a connu un destin similaire. D'un logiciel payant, il est devenu un logiciel gratuit avant d'atteindre le statut d'*open source*. Il continue ainsi à être régulièrement mis à jour, ayant déjà permis à ses utilisateurs de créer plusieurs centaines de jeux de rôle et d'aventure en 2D. Plus récent, nous pouvons mentionner *Open Beats of Rage* (Lavalit Team, 2012), spécialisé dans la création des jeux de combats. Ce logiciel est basé sur *Beats of Rage* (Senile Team, 2003), qui fut abandonné par ses concepteurs au bout de quelques années. Ces derniers ont alors accepté d'en ouvrir le code source, permettant à une communauté de développeurs adeptes de la philosophie du Libre de continuer à le faire évoluer.

Bien qu'intéressants, ces exemples d'usines à jeux distribuées sous forme de logiciels libres sont une exception : les outils destinés à faciliter la création de jeu vidéo par des amateurs sont généralement des logiciels propriétaires². Mais si leurs concepteurs ne sont pas forcément adeptes

1. Acronyme de American Standard Code for Information Interchange. Il s'agit d'une norme informatique pour l'encodage des caractères. Avant l'apparition de moteurs graphiques performants, il était possible de détourner les fonctions d'affichage textuel d'un ordinateur pour simuler des dessins, en utilisant notamment les caractères spéciaux définis dans la norme ASCII.

2. Sur les 452 logiciels référencés sur la base de données <http://creatools.gameclassification.com>, seuls 81 outils sont des logiciels libres, soit environ 18 % des usines à jeux référencées sur cette base de données.



Official Hamster Republic Role Playing Game Creation Engine est une usine à jeux qui, en passant du statut de logiciel propriétaire à l'*open source*, a gagné en popularité et en longévité. (Par Bob the Hamster, GNU GPL)

du Libre, la situation semble être différente pour certains de leurs utilisateurs. À la manière des « clubs » qui permettaient l'échange de jeux amateurs sous forme de code source, de nombreux utilisateurs d'usines à jeux se structurent en « communautés de créateurs » dédiées à un logiciel donné par le biais d'Internet. Par exemple, le site *The Daily Click* (2002-2012) est dédié aux usines à jeux créés par la société Clickteam, tandis que *Game Maker Games* (2004-2012) est centré sur les différentes versions de *Game Maker*. Si les jeux échangés sur ces sites sont généralement « protégés » afin que leur code source ne soit pas accessible, nombre d'entre eux sont également diffusés sous forme *open source*. Cela permet d'aider les débutants dans leur apprentissage, ou encore de donner la possibilité aux membres d'une communauté de tenter d'améliorer un jeu vidéo créé par d'autres amateurs. Cette pratique informelle et spontanée de la part des utilisateurs existe depuis *Eamon*, grâce à des communautés basées sur des serveurs BBS ou des sites Internet. Cependant, certains créateurs d'usines à jeux ont également tenté de légitimer et d'encourager cette pratique à travers le courant du *Jeu 2.0*.

3.3 Le « Jeu 2.0 »

Nous pourrions définir simplement le *Jeu 2.0* comme la rencontre entre des logiciels de création vidéoludique existants depuis plusieurs années, les usines à jeux, avec des plates-formes de partage inspirées du *Web 2.0*. Certes, les créateurs de jeux vidéo amateurs n'ont pas attendu l'arrivée du *Jeu 2.0* pour partager leurs créations. Comme évoqué dans la section précédente, de nombreux sites Internet permettent de diffuser des jeux vidéo amateurs. Mais dans tous ces exemples, les plates-formes de partage ne sont pas intégrées au logiciel de création. Il faut quitter ce dernier afin d'échanger et/ou de jouer. L'apport du *Jeu 2.0* ne se situe donc pas du côté des capacités de création mais au niveau du « partage », à travers la mise à disposition non pas d'un logiciel, mais d'une véritable plate-forme intégrée qui associe outil de création et fonctionnalités communautaires. Au sein d'une application unique, il est dorénavant possible de créer, d'échanger et de jouer à du « contenu ludique ».

Le courant du *Jeu 2.0* vise ainsi à encourager les expériences collaboratives de création vidéoludique. En facilitant les échanges entre créateurs, un même projet peut être conçu par plusieurs personnes différentes. Par exemple, avec *The Sims Carnival* (Electronic Arts, 2008), tout internaute peut, en cliquant sur un bouton, modifier un jeu existant sur la plate-forme de partage et en publier une variante. Cette plate-forme gardant une trace des contributions de chacun, il est possible de voir des créateurs qui ne se connaissent pas œuvrer collectivement à la réalisation d'un même jeu vidéo. Pour cela, il suffit que chaque créateur travaille à tour de rôle sur la version créée par un des participants au projet. Dans un registre similaire, chaque partie du jeu *Spore* (Electronic Arts, 2008) contient des éléments qui ont été créés par différents joueurs. Lorsqu'un joueur lance une nouvelle partie, le jeu va récupérer sur la plate-forme d'échange des objets imaginés par d'autres joueurs, tels que des créatures ou des bâtiments, et les utiliser pour construire un monde unique. S'il le souhaite, le joueur peut même modifier ces créations provenant d'autres joueurs, avant de les partager à nouveau.

Cependant, cette association profonde entre un logiciel de création vidéoludique et une plate-forme de partage n'est pas sans conséquence. En effet, si l'éditeur du logiciel « débranche » la plate-forme de partage, l'ou-

til de création disparaît avec elle. Or, faire vivre une plate-forme de partage représente un coût financier non négligeable. Nous comprenons que pour les *Jeux 2.0* du secteur du divertissement, la concurrence est très rude. De nombreux outils risquent de disparaître s'ils ne rencontrent pas le succès escompté par leur éditeur. Lors d'une étude sur le *Jeu 2.0*¹, nous avons identifié 31 outils de ce type qui ont été publiés entre 2007 et 2011. Aujourd'hui, 10 de ces outils ont disparu suite à la fermeture de leur plate-forme de partage par leurs éditeurs respectifs. L'approche du *Jeu 2.0*, qui vise à pousser les utilisateurs d'usines à jeux à partager leurs créations selon des modalités reprenant une partie de la philosophie du Libre, est des plus intéressantes. Pourtant, il semble qu'un modèle économique adapté reste à trouver. En effet, les plates-formes de partage sont pour l'instant toutes fournies gracieusement en dépit de leur évident coût de fonctionnement. Cela explique que de nombreux outils disparaissent, faute de pouvoir être financés à perte par leur éditeur. Dans l'idéal, un meilleur modèle économique devrait donc permettre aux plates-formes de *Jeu 2.0* de subsister, voire d'être profitables, sans pour autant empiéter sur la liberté de création qu'elles offrent aux utilisateurs. C'est un défi auquel ont fait face les sites du *Web 2.0*, à l'image de Wikipédia dont la pérennité repose en grande partie sur les dons qu'elle reçoit, ou de Youtube qui s'appuie sur la diffusion de publicités. L'exemple du *modding* nous montre que l'industrie des jeux vidéo a déjà réussi, par le passé, à imaginer un modèle économique permettant de faire cohabiter son fonctionnement profondément propriétaire avec la philosophie du Libre. L'histoire se répètera-t-elle pour le *Jeu 2.0*? Face au formidable potentiel de création et de partage que ce courant représente pour les créateurs amateurs de jeux vidéo, nous ne pouvons que l'espérer...

Conclusion

Depuis ses premiers jours, les profits de l'industrie du jeu vidéo reposent avant tout sur la commercialisation de logiciels propriétaires dont l'utilisation et la diffusion sont artificiellement encadrées afin d'assurer une

1. Damien DJAOUTI, « Serious Game Design - Considérations théoriques et techniques sur la création de jeux vidéo à vocation utilitaire », thèse de doct., Toulouse : Université de Toulouse, 2011, URL : http://www.ludoscience.com/files/these_djaouti.pdf.

rentabilité maximale. Pourtant, les jeux vidéo ne sont pas étrangers à la philosophie du Libre, comme nous le montrent les trois périodes historiques abordées dans ce chapitre.

Tout d'abord, l'invention même des jeux vidéo s'est faite dans le cadre de logiciels au code source ouvert et à la distribution sans entraves. Ces approches seront ensuite formalisées par les courants du logiciel libre et de l'*open source*. La création du premier jeu vidéo ayant eu une influence notable, *Spacewar!* (1962), doit d'ailleurs beaucoup à son statut ouvert, qui préfigure celui des logiciels libres. Comme nous l'avons évoqué, la première version du jeu n'était pas particulièrement captivante, et souffrait de nombreux défauts de jouabilité. Le code source du jeu étant ouvert, ce dernier a pu évoluer de manière considérable grâce aux ajouts de personnes qui n'étaient pas à l'origine du projet. Grâce à ces améliorations successives, il est devenu particulièrement populaire au sein du MIT. De plus, l'ouverture de son code source s'appliquait également à sa diffusion, voulue aussi libre que possible. Il a donc pu être diffusé vers les diverses universités américaines, permettant à de nombreux étudiants de découvrir les jeux vidéo. Influencés par cette première expérience vidéoludique, et après en avoir étudié le code source, certains étudiants à l'esprit entrepreneur ont alors eu l'idée de les commercialiser en les transformant en logiciels propriétaires au passage.

Pourtant, l'influence de l'esprit du logiciel Libre dans l'histoire du jeu vidéo ne se limite pas à son commencement, même si c'est la période où elle a été la plus forte. Face au verrouillage des circuits de distribution par les plus grosses entreprises de l'industrie vidéoludique, certains créateurs indépendants ont profité de l'apparition de nouveaux modes de communication plus libres, tels que les serveurs BBS et Internet, pour diffuser leurs œuvres. Ces circuits de diffusion parallèles, ouverts par nature, pouvaient alors à nouveau accueillir des jeux *open source*, des jeux propriétaires mais gratuits (*freeware*) ou encore des jeux commerciaux au modèle économique singulier (*shareware*). C'est au sein des jeux vidéo distribués en tant que *shareware*, et plus particulièrement avec *Doom* (1993), que se développe un courant associant une pratique du Libre au logiciel propriétaire de manière originale : le *modding*. Certains créateurs de jeux vidéo commerciaux diffusent sciemment leurs outils de développement afin de permettre à quiconque de créer et diffuser librement des variantes de leur

jeu. En contrepartie, les joueurs souhaitant utiliser ces variantes doivent auparavant acheter le jeu originel. Ce modèle atypique, propre à l'industrie du jeu vidéo, permet d'offrir à tout utilisateur certaines des libertés du logiciel libre (étude, modification, amélioration, distribution) sans pour autant atténuer la rentabilité commerciale du logiciel propriétaire.

Mais l'influence du logiciel libre ne se borne pas à l'aspect industriel du jeu vidéo, puisqu'elle est également notable au sein de la création vidéoludique amateur. La philosophie du Libre était particulièrement présente lors de ses balbutiements, lorsque les créateurs s'échangeaient les codes sources de leurs jeux directement dans des clubs d'utilisateurs, ou les publiaient dans des livres et des magazines. Cette liberté a, par la suite, eu tendance à disparaître, suite à l'arrivée de logiciels permettant aux amateurs de créer facilement des jeux vidéo sans forcément avoir à les programmer : les « usines à jeux ». Nombre d'amateurs ont alors choisi de ne plus diffuser le code source de leurs créations, même si certains ont continué à le faire, notamment pour aider les amateurs débutants. Au-delà de l'apparition notable de quelques « usines à jeux » sous statut *open source* qui leur assure une longévité exceptionnelle, l'influence du Libre sur la création vidéoludique amateur est globalement restée limitée depuis l'arrivée des usines à jeux dans les années 1980. Pourtant, en 2007, est apparu un courant qui tente de ré-insuffler les notions de partage de code source et d'amélioration collaborative de jeux amateurs : le *Jeu 2.0*. En associant des plates-formes de partage héritées du *Web 2.0* avec des « usines à jeux », le *Jeu 2.0* pousse les créateurs amateurs à publier en *open source*. Si l'avenir du *Jeu 2.0* sur le long terme reste encore incertain, du fait de l'absence de modèle économique adéquat, son existence montre que, bien qu'elle ne soit pas aussi forte qu'au début de son histoire, l'influence de la philosophie du Libre reste encore présente dans le monde des jeux vidéo. . .

Comme toute étude historique, cet article n'a pas prétention à l'exhaustivité, et se focalise sciemment sur certains aspects de l'histoire du jeu vidéo, et notamment sur sa vision américaine et européenne. Bien qu'elle puisse par la suite être complétée par l'étude d'autres périodes historiques, ou par une vision consacrée à d'autres régions du monde¹, nous pensons

1. En particulier l'histoire du jeu vidéo au Japon, autre grand marché historique avec les États-Unis et l'Europe.

néanmoins que les trois éléments de cette histoire illustrent globalement l'influence du Libre sur ce secteur. Si la philosophie qui allait devenir celle du Libre n'a vraiment été majoritaire dans le jeu vidéo qu'à ses débuts, son esprit n'a pourtant pas été complètement effacé par les chiffres d'affaires réalisés grâce au modèle propriétaire. De temps à autre, celle-ci réapparaît donc au sein du jeu vidéo, s'associant de différentes manières avec le modèle propriétaire dominant. Mais, plus que le simple partage de code source, ou d'éléments graphiques et sonores, le principal apport de cette influence résiduelle est de permettre à tout joueur de devenir un créateur. Grâce aux différentes technologies de la communication, les joueurs peuvent même partager leurs créations ou modifications au sein de communautés de passionnés. Finalement, si le jeu vidéo facilite bien souvent la construction de lien social entre les joueurs, le Libre leur permet également d'exprimer leur créativité...

Références

- AHL, David H., *BASIC Computer Games : Microcomputer Edition*, Microcomputer ed, New York : Workman Pub Co, 1978.
- BARTON, Matt et Bill LOGUIDICE, « The History of Spacewar ! : The Best Waste of Time in the History of the Universe », dans : *Gamasutra* (2009), URL : <http://www.gamasutra.com/view/feature/4047/>.
- « The History of the Pinball Construction Set : Launching Millions of Creative Possibilities », dans : *Gamasutra* (2010), URL : <http://www.gamasutra.com/view/feature/3923/>.
- CHAPLIN, Heather et Aaron RUBY, *Smartbomb : The Quest for Art, Entertainment, and Big Bucks in the Videogame Revolution*, New York : Algonquin Books, 2005.
- CRAIG, Paul, *Software Piracy Exposed*, Waltham : Syngress, 2005.
- DJAOUTI, Damien, « Les pionniers du jeu vidéo », dans : *Pix'n Love* 11 (2010).
- « Les pionniers du jeu vidéo : Spacewar ! », dans : *Pix'n Love* 12 (2010).
- « Serious Game Design - Considérations théoriques et techniques sur la création de jeux vidéo à vocation utilitaire », thèse de doct., Toulouse : Université de Toulouse, 2011, URL : http://www.ludoscience.com/files/these_djaouti.pdf.

- DONOVAN, Tristan, *Replay : The History of Video Games*, Yellow Ant, 2010.
- GRAETZ, Martin, « The Origin of Spacewar ! », dans : *Creative Computing* 7.8 (1981), p. 56–67.
- KENT, Steven L., *The Ultimate History of Video Games : From Pong to Pokemon—The Story Behind the Craze That Touched Our Lives and Changed the World*, Three Rivers Press, 2001.
- KUNZE, Frank et Greg GIOLA, « A Brief Overview of Eamon », dans : *Eamon Adventurer's Guild Online* (2008), URL : <http://www.eamonag.org/pages/eamonoverview.htm>.
- KUSHNER, David, *Masters of Doom : How Two Guys Created an Empire and Transformed Pop Culture*, Random House Trade Paperbacks, 2004.
- LAUKKANEN, Tero, « Modding Scenes - Introduction to user-created content in computer gaming », dans : *Hypermedia Laboratory Net Series* 9 (2005), URL : <http://tampub.uta.fi/handle/10024/65431>.
- LEARMONTH, Michael, « No Pain, No Game », dans : *Metroactive.com* (1999), URL : <http://www.metroactive.com/papers/metro/09.16.99/cover/bushnell2-9937.html>.
- LEVY, Steven, *Hackers. Heroes of the Computer Revolution*, New York : Dell Publishing, 1994.
- NIEBORG, David B. et Shenja van der GRAAF, « The mod industries? The industrial logic of non-market game production », dans : *European Journal of Cultural Studies* 11.2 (2008), p. 177–195, URL : <http://ecs.sagepub.com/content/11/2/177>.
- PITTS, Bill, « The Galaxy Game », dans : *Stanford Computer History Exhibits* (1997), URL : <http://infolab.stanford.edu/pub/voy/museum/galaxy.html>.
- POSTIGO, Hector, « Of Mods and Modders Chasing Down the Value of Fan-Based Digital Game Modifications », dans : *Games and Culture* 2.4 (2007), p. 300–313, URL : <http://gac.sagepub.com/content/2/4/300>.
- WOLF, Mark J. P., éd., *The Video Game Explosion : A History from PONG to PlayStation and Beyond*, Westport : Greenwood, 2007.

Brève histoire de l'identité visuelle du Libre

Thibaud HULIN

Du panda roux de Firefox à la mouette d'OpenOffice en passant par le manchot de Linux, toutes ces mascottes animalières ont investi nos écrans. Ce « bestiaire exotique » en vigueur dans le monde du libre contraste avec l'esprit de sérieux qui domine le visuel d'autres logiciels qui soignent la référence à notre quotidien matériel : la lettre e de Internet Explorer, la fenêtre de MS Office, la pomme d'Apple... Ces logos ne sont pourtant qu'un des éléments parmi d'autres de l'espace symbolique dans lequel s'inscrivent les choix graphiques et interactifs qui constituent l'identité visuelle d'un logiciel. Raymond (2001) a décrit les spécificités organisationnelles du Libre comme relevant du genre « bazar ». À notre tour, comment décrire les spécificités de l'identité visuelle du Libre, laquelle réunit des registres sémiotiques aux fonctionnements multiples ? Pour saisir puis analyser cette originalité visuelle, nous nous appuyerons sur la définition des logiciels libres, qui présente quatre libertés fondamentales, structurées chacune par un registre de contrainte propre. Nous montrerons que ce jeu de libertés et de contraintes soulève la question de l'histoire de l'identité visuelle, qui est une histoire à la fois technique et humaine.

Pour soutenir notre hypothèse d'une identité visuelle propre au Libre, nous analyserons l'évolution de quelques formes visuelles de projets

libres à l'aide de notre outil liberté / contrainte : Mozilla, KDE, OpenOffice, Ubuntu. Ainsi décrirons-nous une identité visuelle du Libre à partir de l'histoire des développeurs et de leurs interactions avec la machine.

Dans une première section, nous présenterons les quatre libertés et les quatre contraintes qui les structurent. Chacune des quatre sections suivantes analyse ces registres avec un exemple de logiciel libre. Elles présentent sommairement un outil conceptuel d'analyse qui soit opérationnel, puis décrit comment s'articulent contraintes et libertés sous un angle historique.

1. Les quatre registres de libertés et de contraintes

Afin de décrire la grammaire visuelle du Libre, il nous faut tout d'abord savoir ce que nous entendons par logiciel libre au-delà de ce que l'on sait de sa définition juridique. Notre perspective est de décrire cette grammaire à partir du concept de liberté. Ce concept de liberté n'est pas l'opposé de la contrainte, et peut donc s'interpréter du point de vue kantien. En effet pour Kant, la liberté est autonomie (*autos-nomos*), c'est-à-dire qu'elle s'impose à elle-même (*autos*) ses propres normes (*nomos*), au lieu que celles-ci lui soient imposées par une instance extérieure. Ainsi une volonté libre est « une volonté soumise à des lois morales »¹. En ce sens, nous interprétons la liberté du logiciel libre non pas comme une liberté sans frein, mais comme un ensemble de choix fondés sur des valeurs. Sur le plan visuel, Kant considère que c'est le libre jeu des facultés de l'entendement qui assure le plaisir esthétique. Or, un logiciel libre articule à la fois une dimension esthétique (le libre jeu des formes visuelles), une dimension éthique (des valeurs et des normes de comportement), et une dimension technique (le produit et le code source soumis à des standards). Ainsi nous ferons l'hypothèse qu'au cœur de l'identité visuelle du logiciel libre se jouent des allers-retours entre des libertés et des contraintes que nous devons définir.

Quelles sont les libertés propres au logiciel libre ? La Free Software Foundation (FSF) en recense quatre : utiliser, redistribuer, étudier et améliorer librement un logiciel. À quelles contraintes ces libertés font-elles

1. Emmanuel KANT, *Fondements de la métaphysique des moeurs*, Paris : Fernand Nathan, 2004.

donc face ? Dans un environnement médiatisé par l'informatique, nous recensons quatre registres de contraintes : les contraintes liées au support numérique ; les contraintes fonctionnelles, liées au type d'application envisagé ; les contraintes liées au point de vue de l'utilisateur, d'ordre sémiotique, physiologique ou cognitif ; enfin les contraintes liées aux valeurs collectives portées par le projet. Nous nous inspirons ici de la théorie des quatre niveaux de Bouchardon et *al.*¹, lesquels distinguent :

- le niveau théorique du numérique (niveau 1), ce qui est le propre du support numérique par rapport aux autres supports d'écriture comme le papier ;
- le niveau applicatif (niveau 2), les fonctionnalités offertes par le logiciel ;
- le niveau interprétatif (niveau 3), la manière dont nous interprétons les formes que nous percevons ;
- et le niveau politique du numérique (niveau 4)² qui s'articule autour des codes, normes et règles implicites ou explicites au niveau de la société ou d'un collectif.

Bien que nous puissions distinguer ces niveaux par l'analyse, ceux-ci constituent des dimensions inséparables de notre vie numérique. Ainsi, on ne peut pas réduire le logiciel libre à sa définition juridique, comme si le seul registre de contraintes à prendre en compte était une partie du niveau politique (niveau 4). Car les normes et les valeurs du logiciel libre continuent à faire l'objet de débats. Par exemple, Stallman³ considère que la force du logiciel libre réside dans sa capacité à construire une société libre (*free society*). À l'inverse, pour Linus Torvalds⁴, c'est l'intérêt et le défi ludique (*entertainment*) de la solution technique qui fait que celle-ci devient un choix de société. À partir d'un point de vue politique très différent, tous deux constatent que les choix techniques (niveau 2) ne sont pas indépendants des débats publics (niveau 4). À leur tour, lorsque des par-

1. Serge BOUCHARDON et al., « Explorer les possibles de l'écriture multimédia », dans : *Les Enjeux de l'information et de la communication* 2011/1 (2011).

2. Le niveau 4 sera envisagé par la suite par Bruno Bachimont, un des auteurs de l'article cité.

3. Richard M. STALLMAN et Joshua GAY, *Free software, free society : selected essays of Richard M. Stallman*, Paris : Lavoisier, 2002.

4. Linus TORVALD et David DIAMOND, *Il était une fois Linux : L'extraordinaire histoire d'une révolution accidentelle*, Paris : Osman Eyrolles Multimédia, 2001.

Libertés	Contraintes	Concepts pour l'analyse
Étudier	Culture numérique	Énonciation éditoriale
Améliorer	Fonctionnalités techniques	Analyse fonctionnelle
Utiliser	Apprentissage humain	Grammaire visuelle
Redistribuer	Normes sociales	Normes et valeurs

Registres des libertés et des contraintes des logiciels libres.

tisans du *open source*¹ considèrent que « la licence ne doit discriminer personne », ils supposent que la sphère technique n'est pas indépendante de la sphère politique.

Le niveau 3, celui de l'activité d'interprétation des formes visuelles, serait-il indépendant des autres registres de contraintes ? Les utilisateurs sont eux aussi des contributeurs actifs des logiciels libres, dans la mesure où ils sont libres de les utiliser et d'en redistribuer des copies. Pour prendre en compte les besoins et les désirs des usagers, et la manière dont ils interprètent et comprennent les interfaces au niveau sémiotique, il existe de nombreux moyens techniques : les statistiques d'usage, récoltées via le système d'exploitation, le nombre de téléchargements ou d'inscriptions, les systèmes de votes pour la résolution prioritaire de bugs ou pour obtenir de nouvelles fonctionnalités, etc. Le registre sémiotique, celui de l'interprétation humaine, est donc un nécessaire *feed-back* pour le développeur : il n'est pas séparable des registres techniques et politiques.

Enfin, le niveau 1, celui du support numérique, contraint lui aussi la production des logiciels libres, car la facilité avec laquelle nous échangeons des documents numériques ou les calculs qu'il semble autoriser pose des problèmes nouveaux ou spécifiques que nous n'avions pas lorsque nous diffusons des documents semblables sur un support papier.

Ainsi, ces quatre registres de libertés et de contraintes constituent donc un outil d'analyse des visuels offerts par les logiciels libres. Nous en proposons l'analyse dans les quatre sections suivantes.

Nous utilisons l'approche sémiotique qui vise ici à analyser les signes visuels et leur signification, c'est-à-dire à montrer ce que ces signes dé-

1. Bruce PERENS, « The Open Source Definition », dans : *Open Source : Voices from the Open Source Revolution*, sous la dir. de Chris DIBONA, Sam OCKMAN et Marc STONE, Sebastopol, CA : O'Reilly, 1999, p. 171-188, URL : <http://oreilly.com/openbook/opensources/book/index.html>.

notent, ce qu'ils disent ostensiblement, mais aussi ce qu'ils connotent, c'est-à-dire ce qu'ils nous montrent de façon implicite. Nous accorderons une attention particulière au contexte de réception des formes perçues, considérant que celles-ci s'inscrivent dans des situations sociales, historiques, techniques, culturelles et économiques données. Pour décrire chacune de ces relations de contrainte, l'analyse portera sur différentes dimensions du logiciel libre : l'énonciation éditoriale, les fonctionnalités, la grammaire visuelle et les normes sociales. Nous montrerons comment l'articulation d'un registre de liberté à un registre de contraintes permet de décrire le fonctionnement sémiotique de différents logiciels.

2. L'énonciation éditoriale dans la mythologie Mozilla

La liberté d'étudier un logiciel semble sans limite dans la mesure où le code est accessible. Pour autant, cette lecture requiert une culture dont tout le monde ne dispose pas. À la différence d'un document papier, ce qui génère le document numérique ne nous est pas forcément accessible. Une page Web sépare forme et contenu : les différents codes (HTML, CSS, Javascript...) ne se lisent pas de la même façon que le document qui s'affiche dans le navigateur. Étudier le code source d'un logiciel libre suppose donc de disposer de compétences techniques qui prennent racine dans une culture numérique avancée, elle-même inscrite dans un espace culturel plus large. Pour décrire le fonctionnement des contraintes qui structurent la liberté d'étudier un logiciel libre, nous utiliserons le concept d'« énonciation éditoriale » de Souchier et Jeanneret¹, qui vise à éclairer les conditions de l'écriture numérique. Avec l'invention de ce concept, leurs auteurs tentent de décrire les conditions de possibilités des formes de l'énonciation en dépassant certaines dichotomies faciles entre culture et technique, texte et document, lecture et écriture, ancien et nouveau, matériel et immatériel, auteur et lecteur, etc. En effet, le média de l'écriture numérique, l'écran et tout ce qui relève du dispositif de lecture numérique, l'« architexte », structurent la communication numérique. Chaque dispositif d'écriture conditionne donc l'énonciation différemment. Or, avec

1. Emmanuel SOUCHIER et Yves JEANNERET, « L'énonciation éditoriale dans les écrits d'écran », dans : *Communication et langages* 145.1 (2005), 3–15.

Christin¹, nous considérerons que l'écriture ne se réduit pas à la production de textes : le texte est à son origine un graphisme, tandis que l'écriture numérique inclut l'écriture multimédia, articulant ensemble formes graphiques et formes textuelles. Analyser le mode d'énonciation éditoriale des développeurs revient donc à analyser les conditions d'apparitions des signes à l'écran, c'est-à-dire, pour nous, une culture numérique qui apparaît particulièrement lorsque les développeurs affirment leur identité. Avec l'exemple de Mozilla, nous allons dans cette section étudier les logos et les noms par lesquels cette identité est affirmée afin de faire apparaître les conditions culturelles de cette énonciation de soi.

Tout d'abord, remarquons que la culture des développeurs ne se limite pas à sa seule dimension technologique. Elle inclut par exemple une « culture geek », nécessaire pour décrypter les écritures, les formes visuelles et les codes sociaux. Cette culture se reflète jusques et y compris dans tous les documents numériques publics relatifs au projet libre. L'évolution du logo de Firefox, comme écriture visuelle, est intéressante à plus d'un titre, dans la mesure où elle évoque les procédés techniques². Même s'il ne s'agit pas du même plan sémiotique qu'une interface de logiciel, sa conception s'inscrit dans le cadre d'une cohérence graphique et sémantique propre au projet de développement, d'autant que dans le cadre des logiciels libres, les développeurs décident généralement du choix du logo, tandis que les graphistes sont aussi développeurs. Or, à son origine, le code de l'actuel navigateur devient *open source* au moment de l'annonce du projet avorté de Netscape 5 en 1998 (projet Gromit), qui débouchera sur l'arrivée de Mozilla 1.0. La version *stand-alone* du navigateur apparaît avec le projet Phoenix (23/09/2002), rebaptisé Mozilla Firebird (toujours l'oiseau de feu) puis Mozilla Firefox³ pour des raisons de marques⁴.

1. Anne-Marie CHRISTIN, éd., *Histoire de l'écriture : de l'idéogramme au multimédia*, Paris : Flammarion, 2011.

2. Pour une infographie sur l'historique de Mozilla Firefox et son logo, voir <http://www.foxkeh.com/downloads/history/history-foxkeh.pdf> et https://people.mozilla.com/~faaborg/files/20090515-creativeBrief/creativeBrief-i1-wm.png_large.png. Pour voir l'historique des interfaces de Netscape, qui nous intéresse moins ici, voir <http://sillydog.org/narchive/>.

3. Tandis que l'oiseau de feu est une référence explicite au Phoenix, l'expression anglaise « renard de feu » désigne en français le panda roux.

4. Après IBPhoenix, c'est au tour du gestionnaire de base de données Firebird de revendiquer ses droits de marque déposée. Cf. <http://news.cnet.com/2100-7344-5156101.html>.

Lorsque qu'on lit les notes des différentes versions, ainsi que les documents des designers qui ont dessiné les logos, on constate que depuis l'origine non libre du navigateur jusqu'à aujourd'hui, la rhétorique officielle évolue à partir d'un ciblage du public et dans le cadre d'objectifs ostensiblement commerciaux, en direction d'un public de plus en plus large et dans un cadre non lucratif. Certaines valeurs traversent cet héritage, comme la rapidité technique qui a toujours été mise en avant. Cependant, le discours officiel devient moins technique, moins dépendant des situations géographiques ou du profil des utilisateurs. Ainsi le globe terrestre ne doit se référer à aucun pays en particulier¹. En effet, à son origine, le logo de NCSA Mosaic en 1993 présente un globe terrestre où l'on voit clairement l'Amérique. Or, dès 1994, dans le logo de Netscape, le globe se présente à l'aide d'un simple arc et d'un jeu de contrastes, pouvant indiquer le sommet de la terre par-dessus lequel un N apparaît en partie, tel le lever d'un soleil, parfois associé à la roue d'une barre de navire. La rhétorique de la navigation maritime, pointant la précision du dispositif technique, est donc abandonnée pour pointer uniquement sur la fonction du dispositif, qui est d'accéder à des informations de niveau mondial. Le remplacement du N de Netscape par une mascotte animalière introduit dès lors un changement d'identité : l'absence de référence directe à un mot évoque davantage les caractéristiques de l'animal choisi pour mieux évoquer l'identité des acteurs du projet, ce qu'ils sont ou ce à quoi ils prétendent. Or, si les changements de noms apparaissent à l'occasion de changements stratégiques², les choix visuels qui les structurent ne s'appuient pas sur la culture économique mais plutôt la culture technique des développeurs, voire la « culture geek ». Par culture geek nous n'entendons pas une culture de l'a-socialisation, ce qu'évoque initialement ce mot, ou par extension limitée à la connaissance des ordinateurs ; mais une culture qui se développe autour de la connaissance et de l'usage des ordinateurs, laquelle ne se limite donc pas aux machines. Par exemple, la référence à

1. « Generic continents (so as not to show any form of geographic preference) », cf <https://blog.mozilla.org/faaborg/2009/05/15/creative-brief-for-the-new-firefox-icon>.

2. Ceci n'est pas propre à Mozilla ; par exemple la distribution Linux Mandrake, en référence au personnage de BD américain, est devenue Mandriva à la suite d'un rapprochement du français MandrakeSoft avec l'entreprise brésilienne Conectiva, mais aussi parce que Mandrake était déjà un nom déposé par l'américain Hearst Corporation qui éditait justement cette BD.

des livres de science-fiction est une constante dans l'histoire de la culture geek dans le milieu du Libre.

Parmi les références éditoriales à l'identité du projet Mozilla, on trouve toute une lexicologie qui peut prétendre à une étymologie originale. Ainsi, le mot Mozilla (tout comme ChatZilla et assimilés) vient de Godzilla, le lézard de science-fiction japonaise, présent dans nombre de BDs de type *mangas*, puis dans les BDs américaines, les *comics*. Ces mots évoquent aussi l'histoire des entreprises du Web : Mozilla veut dire aussi Moz-kill, ce qui correspond en phonétique à « Mos-killer », le tueur de Mosaic, ce dernier désignant le concurrent de l'équipe Netscape NCSA Mosaic (qui reprenait sans vergogne le nom de son concurrent). Le mot Mozilla aurait été proposé par le développeur de Netscape et futur patron de boîte de nuit Jamie Zawinski, selon ses dires, le 5 août 1994 lors d'une réunion. Le mot s'est imposé progressivement jusqu'à désigner l'ensemble de la suite des logiciels que l'on connaît aujourd'hui.

La référence à des mascottes, à des animaux pas forcément en voie d'extinction, trouve donc ses racines dans une mythologie. Le projet *Phoenix browser*, qui visait à construire un navigateur indépendant (*stand-alone*) est à l'origine du nom Firebird (qui donna Thunderbird et Sunbird), puis de Firefox : le feu fait référence à l'antique Phénix grec. Les noms initiaux des projets de courriel Mozilla Minotaur (autre référence directe à la mythologie grecque) et Phoenix furent abandonnés. On trouve d'autres références mythiques avec le projet Camino (le chemin en espagnol), le navigateur pour Mac OS/X, initialement baptisé projet Chimera en raison du mélange hétéroclite de technologies et en référence aux chimères grecques. Les mythes n'ont pas tous des origines grecques bien sûr. Par exemple, la technologie XUL qui est une technologie d'affichage XML (prononcez *zoul*), fait référence au demi-dieu Zoul dans le film américain « SOS fantômes » (*Ghostbusters*).

Logiciel	Date	Document	Publics	Idees cles	References	Valeurs
Netscape 1.0	13/10/1994	Declaration de presse	Entrepreneurs, utilisateurs individuels et academiques	Navigateur gratuit; commerce	Commerce, concurrence (NCSA Mosaic), <i>open software</i>	Rapidite, simplicite, elegance, innovation
Mozilla 1.0	05/06/2002	Site de bienvenue	Developpeurs, tout utilisateur (testeur)	Code source pouvant etre reutilise et integre	<i>open source</i>	Respect des standards, la vitesse et la portabilite, stabilite
Phoenix 0.1	23/09/2002	Notes de version	-	Personnalisation, confort, vitesse, nombre des caracteristiques	Mozilla, Minotaur, Technologies (XUL)	Minceur, chic, « faire ce qui est juste »
Mozilla Firefox 0.8	14/02/2004	Notes de version	Amis, famille, collegues	Navigation par onglets, blocage des popup, gestionnaire de telechargements.	Mozilla, Phoenix Firebird, Kazaa, IBM	Rapidite, fonctionnalites, efficience, distribution de masse
Mozilla Firefox 1.0	09/11/2004	Notes de version	Amis, famille, collegues	Navigation par onglets, plurilinguisme	Mozilla Thunderbird, Kazaa	Rapidite, fonctionnalites, efficience, distribution de masse
Mozilla Firefox 3.5	06/05/2009	Creative brief	-	Simplifier, moderniser	La Bible (Samson et les Philistins), textures, planete Mozilla	Details, pas de preference geographique, trois dimensions, rapidite

Etapes donnant lieu à l'actuel graphisme de Mozilla Firefox

En fin de compte, les conditions d'énonciation de la rhétorique Mozilla contrastent par rapport au projet propriétaire initial. Avec Netscape Communications, les discours s'orientaient vers des publics plus ou moins ciblés. Avec le montage des équipes de Mozilla, la culture interne des geeks l'emporte, appuyée sur la volonté de construire un imaginaire, une mythologie moderne qui se réfère à la guerre économique entre navigateurs concurrents sur fond de *comics* et de *mangas*. Cette mythologie est une construction volontaire des développeurs et pas seulement un projet marketing. Elle s'appuie sur une idéologie, qui sert le logiciel libre, et représente un ensemble de valeurs et de croyances qui accompagne la production technique. Comme le montre Barthes¹, dans les mythologies modernes, la chaîne du signe signifiant / signifié se double, le signifié devenant le signifiant d'autre chose. Ainsi, les signes-mots Mozilla, Thunderbird, Camino, Seamonkey, etc., renvoient à des logiciels, mais aussi à un ensemble de représentations, d'affirmations et de valeurs qui forment au final une idéologie, une vision de la société qui repose avant tout sur des goûts et des croyances. Elle développe un vocabulaire particulier qui a son histoire voire sa grammaire. Par exemple, il est possible de dériver des noms à l'intérieur de la « Mozilla-langue » : Thunderbird donne lieu à Tinderbox, etc. Ce travail linguistique permet de créer une mythologie propre à Mozilla, même au prix de consensus nécessaires entre la culture des développeurs et le public, et peut contenir plusieurs niveaux de signification. Ainsi, le mot *seamonkey* (le singe de la mer) désigne en anglais le crustacé artémie, mais aussi le successeur de la suite Mozilla. Au niveau mythique, c'est une version « politiquement correcte » de *buttmonkey* (le cul du singe) qui ne désignait pas moins que l'équipe de développement de Netscape 6² ! La multiplication des sens renforce bien sûr la force mythique de tels signes, rendus finalement plus accessibles que la mythologie grecque par la modernité et par l'humour. . .

Ainsi, l'histoire de la culture visuelle et textuelle des développeurs du libre tend à créer une dimension mythique. Or, le rôle de tout mythe est de se référer à un monde pour l'expliquer lorsque la raison n'est plus utile. La référence à un monde propre est explicite lorsque les développeurs parlent de « planète Mozilla ». Celle-ci prend forme dans le logo Fire-

1. Roland BARTHES, *Mythologies*, Paris : Seuil, 1957.

2. Source : https://wiki.mozilla.org/SeaMonkey:Name_And_Version.

fox avec ce globe qui ne fait référence à aucun contexte culturel ou géographique particulier, et présente finalement une « planète de martiens » comme on a pu le lire dans un forum. Au final, cette mythologie diffuse et trouve ses racines dans la culture et les valeurs américaines, en proie à la mondialisation, même donc lorsqu'elle récupère d'autres mythologies, européennes ou japonaises. Le support numérique favorise et conditionne largement cette diffusion, tandis que la culture numérique s'enrichit de nouveaux mythes. Cette mythologie contraste nettement avec la rigueur du développement et le respect logique de la syntaxe du code, le *logos* (la raison en grec). Comme l'ont montré Souchier et Jeanneret, l'analyse de l'énonciation éditoriale fait apparaître une idéologie de la démocratisation du savoir qui contraste avec les codes culturels implicites des « lettrés du numérique ». Ainsi, la liberté d'étudier les logiciels libres nous paraît largement conditionnée par la culture geek comme contrainte structurante qui vise, contre toute attente, à articuler ensemble *muthos* et *logos*.

3. Y a-t-il une ergonomie propre au libre ? Le cas KDE

Après avoir analysé les contraintes culturelles qui structurent la première des quatre libertés du logiciel libre, nous étudierons dans cette section la seconde liberté, celle d'améliorer un logiciel libre, afin de faire apparaître le type de contrainte qui la structure. La liberté d'améliorer un logiciel libre est contrainte par le type d'application envisagé, ce qui renvoie à un ensemble de fonctionnalités attendues. Or, beaucoup d'utilisateurs non avertis considèrent encore aujourd'hui que le Libre est peu *user-friendly*, peu adapté à l'utilisateur lambda. Sans doute beaucoup de projets, réalisés exclusivement par des développeurs polyvalents, ont contribué à cette image. Par exemple, l'usage fréquent de la console dans les systèmes de type Unix par leurs utilisateurs peut dérouter l'utilisateur habitué des interfaces graphiques. En déduire que l'usage de la console ne dérive pas d'un processus ergonomique, c'est oublier que la console peut-être très bien adaptée à l'utilisateur qui en possède les compétences. Ergonomique ne signifie pas « adapté à n'importe qui », mais au contraire « bien adapté à certains utilisateurs ». Aussi, de nombreux projets libres font appel à des ergonomes professionnels dans le cadre d'équipes dédiées qui peuvent produire des guides d'identité visuelle (*community identity*

guidelines), d'ergonomie (*usability / human interface guidelines*) et d'accessibilité (*accessibility guidelines*). C'est le cas par exemple de l'environnement de bureau KDE et de son gestionnaire de fenêtres.

L'intervention d'ergonomes est une garantie censée assurer l'adaptabilité des interfaces aux capacités cognitives humaines. Si nous suivons l'historique des interfaces dans notre figure 1, nous observons que l'évolution va vers une interface simplifiée, moins contrastée, vers des couleurs plus nuancées (présence de dégradés, passage du gris au gris bleuté). En suivant notre « cheminement cognitif »¹ personnel sur les dernières moutures, nous observons que la date devient rapidement plus lisible avec des poignées plus légères. Les indicateurs de fenêtre s'allègent et le graphisme s'adoucit, nous faisant oublier progressivement le gris anthracite en vigueur dans les interfaces des années quatre-vingt-dix. L'ajout d'effets et d'animations rend progressivement l'expérience utilisateur plus conviviale. À partir de la version 4, KDE n'indique plus le bouton de menu (tout à fait à gauche) par un triangle noir, ni les différents bureaux (de un à quatre) par des chiffres par défaut. Enfin, tandis que KDE à l'origine était accompagné d'un nombre impressionnant de logiciels dotés d'interfaces graphiques pour effectuer des réglages très fins du système, le gestionnaire de bureau recentre les possibilités en fonction des besoins courants de l'utilisateur. Il introduit notamment le gestionnaire d'activités (ou d'agencements), qui permet de n'afficher que les fenêtres et logiciels correspondants à l'activité de l'utilisateur. Les besoins du public visés par KDE ont évolué.

Pour autant, cette évolution ne s'est pas toujours faite en douceur. Le gros travail d'ergonomie réalisé pour le passage à la version 4 a entraîné de nombreuses discussions et critiques dans la communauté. Celles-ci ont visé les bugs rencontrés et la disparition de fonctionnalités. Plus étonnant, la ressemblance de l'ensemble avec Windows Vista faisait partie des jugements négatifs, alors que cette ressemblance pourrait être un argument favorable à l'arrivée de nouveaux utilisateurs. Cependant, un travail d'autonomisation graphique par rapport à Windows Vista est visible lors du

1. Cathleen WHARTON et al., « The cognitive walkthrough method : A practitioner's guide », dans : *Usability inspection methods*, sous la dir. de J. NIELSEN et R. MACK, New York : John Wiley et Son, 1994, 105–140.

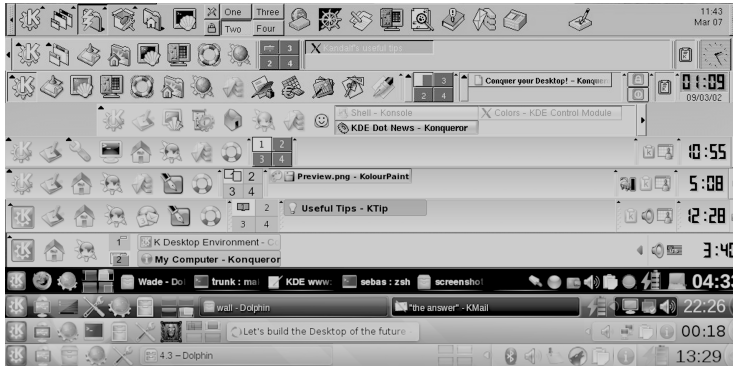
passage de la version 4.0 à la 4.3 (quatre dernières barres sur la figure suivante).

Cette problématique de la différenciation identitaire visuelle se retrouve dans d'autres environnements de bureau, par exemple avec Unity, soutenu par Canonical dans les dernières versions d'Ubuntu, ou de Gnome Shell 3.0. Les dernières versions ont été particulièrement innovantes sur le plan ergonomique, en proposant par exemple un menu d'applications qui s'affiche sur tout l'écran. Le problème de l'évolution des interfaces est alors posé, certains souhaitant largement conserver les bonnes habitudes plutôt que de renouveler l'expérience utilisateur. Dans le cas des logiciels libres, ce problème de l'évolution des interfaces ne peut être envisagé sans prendre en compte d'autres critères que ceux de la pure « expérience utilisateur ». Or, cette expression peut donner à croire que l'utilisateur est un être abstrait, indépendant de son vécu. Au contraire, la démarche ergonomique consiste à prendre en compte l'utilisateur dans sa spécificité, dans ses valeurs et dans son histoire. Dans la mesure où il existe une histoire du logiciel libre, il existe une ergonomie spécifique au libre, qui rend impossible de copier, de transposer ou de s'inspirer fortement d'une interface propriétaire pour l'implanter dans un logiciel libre. Ainsi nous doutons qu'un projet durable de logiciel libre puisse s'envisager à partir de l'idéal rationaliste de la *tabula rasa* dont parlait Descartes, qui peut prendre la forme d'une simple « analyse de l'existant » ou « analyse concurrentielle » et autres analyses de l'environnement extérieur. L'évolution d'un projet doit pouvoir aussi se penser à partir d'une analyse interne, afin de bien comprendre l'histoire fonctionnelle, technique, et donc sociale des communautés concernées, à moins de courir le risque de divisions (*forks*¹) parmi ces communautés.

4. OpenOffice et l'apprentissage de la grammaire visuelle

Dans les deux exemples précédents, nous avons montré que la culture numérique (niveau théorique) ainsi que les fonctionnalités techniques (niveau applicatif) sont des contraintes qui structurent les deux premières

1. Un *fork* est un moment où une partie d'une communauté d'un projet libre se détache du projet initial pour créer un projet différent à partir du même code source. Il existe de nombreux *forks* dans l'histoire des projets libres, par exemple entre OpenOffice et LibreOffice.



Évolution des barres d'outils de l'environnement KDE (la version 4.0 est en noir comme dans MS Vista). Les 8 premières versions utilisent un fond grisé, la suivante est noire, les trois dernières utilisent un fond bleuté sobre.

libertés du logiciel libre, qui sont celles de l'étudier et de l'améliorer. Nous abordons maintenant la troisième liberté, qui est d'utiliser un logiciel libre, en nous concentrant cette fois sur le niveau interprétatif. Ce niveau est le niveau le plus connu du lecteur, puisqu'il s'agit de la manière dont l'utilisateur comprend ce qu'il voit à l'écran sans avoir accès au code qui génère les formes visuelles perceptibles. Aussi, nous précisons en quoi la « grammaire visuelle » sous-tend notre perception et donc nos usages.

La troisième liberté qui est d'utiliser librement un logiciel libre nous semble doublement limitée : par le temps d'apprentissage de l'utilisateur, et par le coût de sa formation (coût de temps, d'énergie et d'argent). En effet, pour comprendre une interface, l'utilisateur doit savoir interpréter le rôle des formes visibles, savoir celles qui sont manipulables, anticiper les réactions, connaître un ensemble de conventions propres au genre de l'application ou au logiciel lui-même, etc. C'est cette activité sémiotique, de perception et d'interprétation simultanée des formes visibles, qui nous permet d'interagir avec un écran. Les développeurs d'interfaces graphiques savent de façon consciente ou non qu'il est possible de réutiliser un ensemble d'objets visuels de façon à profiter de l'expérience de l'utilisateur : c'est le cas des fenêtres, des icônes ou des menus. De même qu'un adepte de la console, ou interface en ligne de commande (*Com-*

mand Line Interface) doit connaître une syntaxe particulière pour pouvoir l'utiliser, il existe un langage informatique visuel, assujéti à une « grammaire visuelle » qui doit être connue de l'utilisateur pour utiliser une interface graphique (*Graphic User Interface*).

Cependant, comme le remarquent Kress et Leeuwen¹, toute grammaire visuelle n'est pas structurée de manière fixe : elle évolue avec le temps, en particulier en fonction des relations sociales qui définissent la manière dont nous interprétons telle ou telle forme. Appliquons cet outil² à l'évolution d'OpenOffice versions 3.2 et 3.3 ainsi qu'au projet intermédiaire Renaissance. L'analyse des métafonctions permet de caractériser les interfaces et de les différencier (Cf. tableau *infra*). Nous reprenons maintenant cette analyse en revenant sur l'histoire de ces versions.

Comme pour le navigateur Mozilla, son histoire est marquée par le départ des développeurs pour travailler à des projets similaires tout en récupérant les sources, de tels *forks* étant un phénomène fréquent dans le monde du libre. Sun Microsystems a racheté la suite propriétaire StarOffice en 1999 et a ouvert le code l'année suivante, donnant lieu à la sortie de OpenOffice 1 en 2002. Oracle Corporation a ensuite racheté Sun en 2010 mais s'est retirée du projet l'année suivante tandis que la communauté créait Libreoffice. Sous la pression d'IBM, le code d'OpenOffice est cédé à la fondation Apache. À l'heure où est édité cet ouvrage, le code de la suite bureautique devrait être déjà fusionné avec celui de la suite concurrente IBM Lotus Symphony³. On trouve d'autres versions basées sur le code d'OpenOffice comme Go-oo ou NéoOffice (adapté au Mac OS/X),

1. Gunther KRESS et Theo van LEEUWEN, *Reading Images : The Grammar of Visual Design*, London : Routledge, 2006.

2. Pour décrire de telles grammaires, ces auteurs étendent le champ de la linguistique fonctionnelle de Halliday (1994) à tous les systèmes de communication sémiotique. Ils distinguent trois métafonctions dans une grammaire visuelle que nous réutilisons dans la table (*infra*) :

- la métafonction idéationnelle, qui relie deux objets de façon relationnelle ou hiérarchique afin de représenter des idées au sujet du monde naturel ;
- la métafonction interpersonnelle, qui manifeste des relations sociales entre le producteur, l'observateur et l'objet représenté ;
- et la métafonction textuelle, qui indique des relations internes à une forme sémiotique, lesquels forment des « textes » : un complexe de signes qui adhèrent entre eux et au contexte, et pour lequel l'ordre ou l'arrangement est tout à fait signifiant.

3. Développé principalement en Chine, IBM Lotus Symphony n'était pas initialement un logiciel libre, mais un *freeware*, donc un logiciel gratuit dont le code n'était pas ouvert.

OpenOffice4Kids (version pour les enfants), etc. Or, hormis pour cette dernière, toutes ces versions ont sensiblement une interface similaire, à la différence de MS Office par exemple qui ne connaît aucun *fork* mais a profondément remanié son interface.

Bien que le calendrier n'ait pas été tenu, il existait un projet de renouvellement de l'interface important chez OpenOffice, le projet Impress Prototype Renaissance de l'équipe UX OpenOffice¹. Une démonstration de la future interface du logiciel de présentation pour Impress 3.3 a même été publiée en ligne en juillet 2009. La comparaison des interfaces précédentes (3.2) et futures (3.3) au projet Renaissance, comme sur la figure 2, permet de circonscrire le projet. Il ressort de l'analyse des documents de travail et de la grammaire visuelle des interfaces que le projet Renaissance a tenté d'initier une rupture visuelle à partir d'un projet centré sur le service à rendre à l'utilisateur, en privilégiant l'efficacité². À la suite d'un comparatif réalisé par l'équipe UX au sujet des concurrents propriétaires d'OpenOffice, les designers semblent séduits par plusieurs points de l'ergonomie de MS Office 2007, en reproduisant des formes similaires comme au niveau de la barre d'outils. Pourquoi le projet Renaissance n'est-il pas mis en œuvre ? D'une part les interfaces de MS Office sont protégées par des brevets, rendant toute inspiration risquée sur le plan juridique. D'autre part, les services autour d'OpenOffice peuvent être vendus plus facilement dans la mesure où l'interface d'OpenOffice est plus proche des perceptions des utilisateurs, habitués à MS Office 2000. Or, tandis que MS Office 2007 a renouvelé son interface, la possibilité pour OpenOffice de se rallier aux habitudes des utilisateurs est un argument pour réduire les coûts de formation. Enfin, on notera bien sûr les changements stratégiques des sociétés impliquées qui jouent un rôle sur les décisions à prendre : le projet Renaissance peut être arrêté pour des raisons de financement liés à une restructuration des services ou à la révision des objectifs stratégiques, etc.

En érudant le projet Renaissance, le passage de la version 3.2 à la version 3.3 introduit un changement d'objectif. Impress 3.2 proposait un outil complet pour soutenir la communication orale pour tous, dans une rela-

1. <http://wiki.openoffice.org/wiki/Renaissance>.

2. À la différence de l'efficacité, l'efficacité vise à réduire le nombre d'interactions et de formes visuelles pour un même résultat.

	Modèles idéationnels	Relations interpersonnelles	Agencement textuel
OpenOffice Impress 3.2	Créativité : dire, commenter	Échange	Potentiel
Prototype Renaissance	Table rase et conviviale	Service	Efficienc
Apache OpenOffice 3.3	Modèle économique, représentation mathématique	Partenariat et entrepreneuriat	Puissance

Analyse métafonctionnelle de l'évolution d'OpenOffice

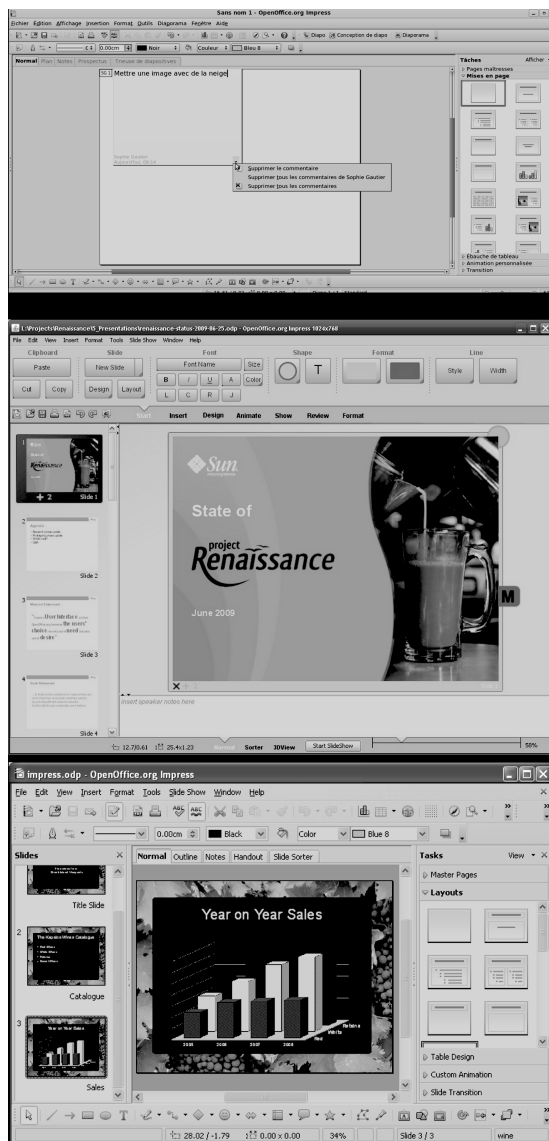
tion d'échange créatif entre utilisateurs et développeurs. Apache OpenOffice 3.3 se veut un outil efficace voire puissant, tourné notamment vers le partenariat et la communication d'entreprise (présentation chiffrée, résultats et ventes, etc.). Le projet Renaissance, malgré ses qualités ergonomiques, semble moins prendre en compte l'historique des organisations impliquées sur le projet OpenOffice en prétendant, comme son nom l'indique, procéder à un renouveau radical ¹.

Ainsi, l'apprentissage de la grammaire visuelle par l'utilisateur dépend de conventions sémiotiques, mais aussi de l'histoire technique : du développement et de l'histoire des organisations soumises à des impératifs économiques. La liberté d'utiliser un logiciel libre est donc contrainte par le dispositif socio-économique dans lequel est inséré le développeur. À leur tour, les usagers interprètent différemment les interfaces en fonction de leur propre contexte socio-économique, d'apprentissage, culturel, de loisir, etc. Un projet ergonomique libre reste bien sûr possible, mais il doit fortement tenir compte de l'histoire économique et sociale de la communauté des développeurs qui oriente les usages.

5. Comment Ubuntu ordonne-t-il le « fatras sémiotique » de ses utilisateurs

Après avoir étudié les trois premières libertés du logiciel libre, nous abordons maintenant la dernière, la redistribution. Nous tentons ici de sai-

1. On pourrait ici comparer le projet Renaissance au Phoenix, qui introduit bien une renaissance dans l'histoire de Mozilla ; or les conditions sont en fin de compte bien différentes, le projet OpenOffice n'ayant pas connu les mêmes déboires et attentes que chez Mozilla, et n'ayant pas forcément besoin de renaître.



Interfaces d'OpenOffice Impress 3.2, le projet Renaissance (non abouti) et Apache OpenOffice Impress 3.3.

sir ce qui contraint, et donc structure, la redistribution. Nous nous plaçons au « niveau politique » du numérique en étudiant ce en quoi les normes sociales contraignent les formes visuelles. Nous prenons comme cas d'analyse le bureau d'Ubuntu en raison de son ambition démocratique clairement affirmée.

Despres-Lonnet¹ qualifie les « écrits d'écran »² de « fatras sémiotique » dans le cas des icônes de bureaux informatiques : les lecteurs, ne parvenant pas à interpréter spontanément les « signes passeurs »³ que sont parmi d'autres les icônes de bureau, sont conduits à mettre en œuvre des stratégies mnémotechniques : ils retiennent l'usage lié à tel symbole plutôt qu'ils ne décodent le symbole à chaque fois qu'ils le voient⁴. Si ce bazar peut désorienter, il est par défaut extrêmement ordonné par un ensemble de règles et de normes qui trouvent leur inspiration dans un projet de diffusion général. Face à la multiplicité des systèmes d'exploitation qui permettent de faire des tâches similaires, le choix de l'utilisateur peut inclure des critères non techniques voire politiques. Dès lors, comment des normes sociales ordonnent-elles le fatras sémiotique de nos bureaux ? Compte tenu de la place dont nous disposons, nous ne proposerons pas ici une étude de chacun des signes passeurs que sont une icône ou une zone cliquable. Nous nous concentrerons sur les changements visuels globaux des versions du bureau par défaut d'Ubuntu comme dispositif symbolique articulant des formes, des sons, des couleurs, des interactions, etc. Nous abordons ici le dispositif Ubuntu (un lexique, des interactions, des formes visuelles, des couleurs, des normes, des règles, des valeurs...) comme « signe passeur global » : à côté de la fonction instrumentale de ce qui se donne à voir ou à lire, il existe une fonction sémiotique qui suscite l'interprétation du lecteur pour anticiper ce vers quoi l'amène une nouvelle version. Nous nous situons donc sur le plan symbolique et social.

1. Marie DESPRES-LONNET, « Écrits d'écran : le fatras sémiotique », dans : *Communication et langages* 142.1 (2004), p. 33-42.

2. L'écrit d'écran, notion initialement proposée par E. Souchier et Y. Jeanneret, définit la sémiotique de l'organisation des signes sur l'écran. SOUCHIER et JEANNERET, *op. cit.*

3. La notion de signe passeur pourrait remplacer celle de lien hypertexte dans la mesure où elle montre aussi la fonction instrumentale du lien, mais montre davantage le fait que la forme visuelle, le nœud, suppose que le lecteur interprète la cible du lien à partir de ce qui est donné à voir.

4. Une disquette continue de signifier « enregistrer », bien que la disquette n'a jamais été manipulée par les jeunes générations.

La liberté de redistribuer des logiciels libres inscrit délibérément le projet technique du libre dans le cadre d'un projet tourné vers la société. À ce titre, le projet Ubuntu ambitionne de distribuer un système d'exploitation à grande échelle. Avec cette ambition, Ubuntu est capable de transformer ses valeurs en normes, de définir ce qui est normal dans un projet libre. Lorsque le projet Ubuntu démarre en 2004 à partir d'une *fork* de la distribution de Linux Debian, il a pour objectif de créer un système d'exploitation accessible au plus grand nombre, ce qui aurait pour conséquence de « démocratiser Linux » ou de produire une « distribution générique »¹. Le projet Ubuntu propose alors par défaut l'environnement Gnome comme bureau. Il est porté par le jeune entrepreneur sud-africain milliardaire Mark Shuttleworth via la société Canonical, à l'aide d'un apport initial de 10 millions de dollars dans la fondation Ubuntu. En référence à ses origines, l'entrepreneur a choisi un nom africain : Ubuntu signifie en bantou « humanité aux autres », avec l'idée de partage et de solidarité (« Je suis ce que je suis grâce à ce que nous sommes tous »). Le projet économique qui accompagne Ubuntu rejoint sa « philosophie », qui se réfère à la promotion du logiciel libre et à l'*open source*, mais aussi au multilinguisme, à la gratuité, à la non-discrimination des personnes. Des expressions comme « tout utilisateur devrait » ou l'idée de « reconnaissance mondiale » montrent que l'ambition d'Ubuntu est de type universaliste². Cette ambition cherche, à partir d'une vision technique, à promouvoir une certaine forme de société. Le passage des valeurs aux normes apparaît avec le projet politique d'Ubuntu, lorsqu'il définit un « code de conduite » ainsi qu'une « politique de gouvernance » pour sa communauté. Dans ces documents, Ubuntu cadre le vivre ensemble à partir de valeurs comme la prévenance, le respect, la collaboration, la consultation, la responsabilité, etc. Ces valeurs, érigées en normes, fournissent donc un cadre politique à l'idée de libre distribution, lequel permet de dire quel

1. Pour une discussion sur les conséquences de cette ambition sur les autres distributions, voir le texte d'Adrian Kingsley-Hughes - 02 avril 2008 - ZDNet.com, traduit en français sur le framablog <http://www.framablog.org/index.php/post/2008/04/06/distribution-ubuntu-substitution-linux>.

2. À la différence du globe présent dans certains logos de Mozilla pouvant évoquer une « planète martienne », le logo d'Ubuntu ne représente pas la terre mais l'humain solidaire ; certaines photographies représentent des personnes pouvant être issues de continents différents. Il n'y a donc pas d'univers Ubuntu mais bien une ambition universaliste liée à un projet économique à l'échelle mondiale.

comportement est jugé normal, et quel comportement est soumis à sanction. L'idée de « communauté » montre bien que si une personne ne se conforme pas aux règles avancées, elle puisse être sanctionnée *a minima* par la non-appartenance à cette communauté. Enfin, ces normes dépassent la sphère de la communauté puisque la « méritocratie ouverte » prônée par Ubuntu constitue une invitation adressée à « tout le monde de n'importe quelle entreprise. »¹

Ces choix éthiques ou politiques s'inscrivent aussi dans les interfaces de la distribution, et notamment le bureau. Par exemple, les choix de palettes de couleurs connaissent une évolution certaine. La première version d'Ubuntu assume nettement sa différence d'avec les autres distributions de Linux, avec un fond de bureau couleur châtaigne, qui connote la terre ou le bois, une certaine authenticité. Les versions suivantes dérivent vers l'orange, en particulier au niveau des icônes, ce qui renvoie à plus de modernité et de convivialité. Le violet des dernières versions, au moment de l'arrivée de la nouvelle interface Unity (Ubuntu 10.04), est lié dans la culture occidentale à l'autorité des évêques et au rite d'initiation : ce choix peut s'expliquer par un compromis entre les couleurs initiales et une volonté de devenir une référence sur un marché des systèmes d'exploitation concurrentiel. L'inversion vidéo des barres de menu, comme lors de la sortie de Windows Vista, se conforme avec la diffusion en masse d'ordinateurs noirs, ce qui à l'origine a surtout pour but d'éviter les traces de doigts. Les bandes noires peuvent aussi renvoyer au cinéma, à son cadre intimiste, tout en nous donnant l'impression d'assister à une histoire extraordinaire, en connotant force et confidentialité. Dans une certaine mesure, l'évolution des chartes graphiques effectue un contrepoint par rapport à d'autres dimensions sémiotiques de la distribution. Par exemple, les noms des différentes versions d'Ubuntu correspondent à des animaux dans la filiation du bestiaire exotique du libre. Ces animaux sont un peu des « signes passeurs » puisqu'ils annoncent une nouvelle version, tout en nous invitant à les interpréter à partir des qualités que nous prêterions à ces animaux. Nous proposons dans le tableau suivant une interprétation de ces signes : noms, couleurs, formes iconiques, menus, etc. Les adjectifs sont particulièrement évocateurs. Par là, les premières versions renvoient à l'idée de recherche d'un positionnement, à de l'auto-dérision

1. <http://www.ubuntu.com/project/about-ubuntu/conduct>.

(verruqueux, vénérable, jovial, pimpant, nerveuse), puis à la recherche de reconnaissance (téméraire, fougueux, robuste, intrépide, enjoué). Les versions suivantes renvoient davantage aux qualités marketing d'un produit à distribuer (karmique, lucide, non-conformiste, chic, onirique, précis).

Ainsi, la diversité et la richesse du libre ne consistent pas à faire une interprétation directe des valeurs du libre. Sur le plan des couleurs, notons qu'à son origine, le marron était censé évoquer l'humanité ou la chaleur¹. Le marron chaleureux de la terre contraste avec le bleu froid du ciel de certains de ses concurrents, tandis que le son du djembé au démarrage nous renvoie à des clichés de l'imaginaire africain occidental. L'histoire du bureau ne montre qu'un ensemble de dérivés par rapport à cette première idée. Les valeurs du libre s'expriment donc à travers les interfaces graphiques tout au long d'une histoire collective propre au projet. Pour les développeurs d'Ubuntu, « notre philosophie est reflétée dans les programmes que nous produisons et incluons dans notre distribution. »²

1. Cf. l'explication que donne M. Shuttleworth sur le wiki d'Ubuntu https://wiki.ubuntu.com/MarkShuttleworth#Why_is_the_default_desktop_in_Ubuntu_BROWN.3F.

2. <http://www.ubuntu.com/project/about-ubuntu/our-philosophy>.

Date	Version	Nom de code	Changements visuels	Effet	Qualificatif	Mascotte
2004-10-20	4.10	Warty Warthog	Couleurs marrons dominantes	Authenticité	verruqueux	phacochère
2005-04-08	5.04	Hoary Hedgehog	Bureau plus lumineux	Performance	vénéérable	hérisson
2005-10-13	5.10	Breezy Badger	Le logo d'Ubuntu devient orange et remplace celui de Gnome	Affirmation	jovial	blaireau
2006-06-01	6.06 LTS	Dapper Drake	Thème orangé	Modérité	pimpant	canard
2006-10-26	6.10	Edgy Eft	Réduction du nombre d'écrans de bureau, l'icône de Firefox remplace l'icône standard; apparition de l'icône d'aide	Simplicité	nerveuse	salamandre
2007-04-19	7.04	Feisty Fawn	L'icône du courriel se simplifie, arrivée de nouveaux appels	Perfection	téméraire	faon
2007-10-18	7.10	Gutsy Gibbon	Applet comportant le nom de l'utilisateur	Personnalisation	fougueux	gibbon
2008-04-24	8.04 LTS	Hardy Heron	Le nom est remplacé par l'icône d'un personnage en vert	Optimisme	robuste	héron
2008-10-30	8.10	Intrepid Ibx	L'icône verte est remplacée par un bouton de mariage rouge avec le nom	Technicité	intrépide	bouquetin

Histoire du bureau d'Ubuntu (1/2)

Date	Version	Nom de code	Changements visuels	Effet	Qualificatif	Mascotte
2009-04-23	9.04	Jaunty Jackalope	Pas de différence visuelle notable		enjoué	jackalope
2009-10-29	9.10	Karmic Koala	Retour à un thème plus sobre ; les icônes des applications passent en noir et blanc, apparition d'une icône de courrier	Sobriété	karmique	koala
2010-04-29	10.04 LTS	Lucid Lynx	Les couleurs vidéos des barres d'outils s'inversent (texte gris sur fond noir), retour de plusieurs écrans du bureau ; disparition de l'icône d'aide	Évasion	lucide	lynx
2010-10-10	10.10	Maverick Meerkat	Apparition d'un thème violet, usage des services sociaux sur le bureau (Gwibber)	Leader	non-conformiste	suricate
2011-04-28	11.04	Natty Narwhal	L'interface Unity remplace celle de Gnome	Ergonomie	chic	narval
2011-10-13	11.10	Oneiric Ocelot	Révision du menu latéral, le logo d'Ubuntu s'agrandit en devenant une icône de ce menu aux côtés de LibreOffice	Affirmation	onirique	ocelot
2012-04-26	12.04 LTS	Precise Pangolin	Retour du nom d'utilisateur en haut à gauche, réduction du nombre d'applications dans le menu latéral	Précision	précis	pangolin

Histoire du bureau d'Ubuntu (2/2)

Références

- BARTHES, Roland, *Mythologies*, Paris : Seuil, 1957.
- BOUCHARDON, Serge et al., « Explorer les possibles de l'écriture multimédia », dans : *Les Enjeux de l'information et de la communication* 2011/1 (2011).
- CHRISTIN, Anne-Marie, éd., *Histoire de l'écriture : de l'idéogramme au multimédia*, Paris : Flammarion, 2011.
- DESPRES-LONNET, Marie, « Écrits d'écran : le fatras sémiotique », dans : *Communication et langages* 142.1 (2004), p. 33–42.
- KANT, Emmanuel, *Fondements de la métaphysique des moeurs*, Paris : Fernand Nathan, 2004.
- KRESS, Gunther et Theo van LEEUWEN, *Reading Images : The Grammar of Visual Design*, London : Routledge, 2006.
- PERENS, Bruce, « The Open Source Definition », dans : *Open Source : Voices from the Open Source Revolution*, sous la dir. de Chris DIBONA, Sam OCKMAN et Marc STONE, Sebastopol, CA : O'Reilly, 1999, p. 171–188, URL : <http://oreilly.com/openbook/opensources/book/index.html>.
- SOUCHIER, Emmanuel et Yves JEANNERET, « L'énonciation éditoriale dans les écrits d'écran », dans : *Communication et langages* 145.1 (2005), 3–15.
- STALLMAN, Richard M. et Joshua GAY, *Free software, free society : selected essays of Richard M. Stallman*, Paris : Lavoisier, 2002.
- TORVALD, Linus et David DIAMOND, *Il était une fois Linux : L'extraordinaire histoire d'une révolution accidentelle*, Paris : Osman Eyrolles Multimédia, 2001.
- WHARTON, Cathleen et al., « The cognitive walkthrough method : A practitioner's guide », dans : *Usability inspection methods*, sous la dir. de J. NIELSEN et R. MACK, New York : John Wiley et Son, 1994, 105–140.

Free software et open source : utopie et idéologie

Sébastien BROCA

La « grande famille » du logiciel libre présente la plupart du temps un front uni vis-à-vis de l'extérieur, et se retrouve sur le principal : la volonté de triompher du logiciel propriétaire. Comme toutes les familles, elle est toutefois sujette à des disputes volcaniques et traversée par quelques inimitiés spectaculaires. Par-delà les conflits individuels, la fracture la plus marquante est celle qui sépare, depuis la fin des années 1990, partisans du *free software* et de l'*open source*.

Cette opposition est un héritage de la création en 1998 du label *open source*, dissidence interne au monde du logiciel libre, impulsée par des informaticiens désireux de rompre avec le discours militant jusqu'alors caractéristique du mouvement. Les créateurs de l'*Open Source Initiative* se sont ainsi efforcés de séduire le monde des affaires, moyennant une nette prise de distance avec l'engagement libriste historique, celui de Richard Stallman et de la *Free Software Foundation*. Les terminologies *open source* et *free software* renvoient depuis lors à deux postures distinctes quant à l'identité et à la finalité du mouvement du logiciel libre. D'un côté, les partisans du *free software* se considèrent comme membres d'un mouvement social, fondé sur la défense intransigeante de certaines valeurs. De l'autre, les partisans de l'*open source* ont promu le logiciel libre

en tant que méthodologie de développement, en insistant sur l'efficacité de celle-ci pour produire des biens informationnels complexes.

Ces divergences d'ordre « philosophique » – comme aiment à les caractériser de nombreux programmeurs – n'empêchent pas les uns et les autres de travailler régulièrement sur des projets communs. Les projets de logiciels libres les plus ambitieux réunissent ainsi des développeurs des deux bords, sans compter ceux qui n'ont pas d'avis tranché ou estiment ce débat dénué de pertinence. La controverse entre *free software* et *open source* n'est toutefois pas sans importance. Elle en dit beaucoup sur les choix stratégiques auxquels se trouve confronté le monde du logiciel libre, aujourd'hui encore. Elle est signifiante, dès lors que le Libre ne concerne plus uniquement le logiciel mais de très nombreux domaines : création culturelle (*Creative Commons* et *Art Libre*), publications scientifiques (*Open Access*), données (*Open Data*), etc. En effet, si le mouvement du logiciel libre esquisse désormais une alternative sociale de portée générale (en termes de modes de collaboration, de rapport au savoir, et de réforme de la propriété intellectuelle à l'ère d'Internet), il n'est pas superflu de s'interroger sur les principes et les objectifs qui doivent, ou devraient, le guider. C'est à cette réflexion que l'examen de la controverse entre *free software* et *open source* – par-delà son intérêt « historique » – peut sans doute contribuer.

1. Le mariage de raison entre GNU et Linux

Afin de comprendre les termes du débat, il est nécessaire de retracer brièvement la trajectoire du logiciel libre dans les années 1990. Au début de la décennie, le projet GNU initié par Richard Stallman en 1984 a produit de nombreux logiciels, mais une composante cruciale du système demeure manquante : le noyau. Le noyau est, comme son nom l'indique, l'élément central d'un système d'exploitation. Il gère les ressources de l'ordinateur et fournit l'interface qui permet aux différents logiciels de communiquer entre eux. Richard Stallman avait très tôt réfléchi à la meilleure manière d'écrire un tel logiciel. Mais cette partie du travail avait été plusieurs fois repoussée, la plupart des participants au projet GNU étant plus enclins à programmer des outils orientés développeurs, comme Emacs. De plus, Richard Stallman pensait adapter un logiciel préexistant,

mais il lui fut pendant longtemps bien difficile de trouver un noyau susceptible d'être réutilisé. Ce ne fut qu'en 1990, lorsque l'université Carnegie Mellon « libéra » le noyau Mach, que le travail sur le noyau GNU put débiter. Lentement.

C'est donc d'une autre manière – de l'extérieur et de façon inattendue – que le système GNU fut complété. Au début de l'année 1991, un jeune étudiant finlandais nommé Linus Torvalds commença à développer un noyau pour Minix¹, afin de pouvoir accéder à son compte Unix depuis l'ordinateur de son domicile, et ne pas être obligé de se rendre à l'université lorsqu'il désirait se connecter. Il rendit rapidement son travail disponible sous licence libre : une licence interdisant toute utilisation commerciale dans un premier temps, puis la *General Public License* à partir de février 1992. Le logiciel fut baptisé Linux, du nom de son créateur, et conformément à la tradition voulant que toute variante d'Unix finisse par la lettre x. Grâce aux listes de diffusion et aux forums électroniques, des développeurs en nombre croissant apportèrent peu à peu des améliorations à la trame réalisée par Linus Torvalds. Le projet prit ainsi une ampleur que celui-ci avait été loin de soupçonner au départ, lorsqu'il parlait de son travail comme d'un « passe-temps [...] pas aussi sérieux et professionnel que GNU »².

En 1994, cette collaboration d'une multitude d'informaticiens à travers le monde avait donné le jour à la version 1.0 du noyau Linux, qui fonctionnait correctement avec l'ensemble du système GNU. Le problème représenté par l'absence de noyau paraissait donc enfin réglé. Un système d'exploitation complet (GNU/Linux) était opérationnel, et Linux était devenu bien plus qu'un substitut provisoire au noyau toujours en développement dans le giron de la *Free Software Foundation*. En effet, le projet de la FSF (Hurd), conçu autour du micro-noyau Mach, s'était entre-temps

1. Minix était un système d'exploitation créé par un professeur néerlandais, Andrew Tanenbaum, à des fins pédagogiques, pour que ses étudiants puissent le manipuler et en comprendre le fonctionnement en un semestre. Il s'agissait d'une sorte de modèle réduit d'Unix. Il avait été entièrement écrit par Andrew Tanenbaum, mais son architecture était semblable à celle d'Unix, de sorte qu'il puisse permettre aux étudiants de se familiariser avec ce dernier. Il s'agissait d'un logiciel développé sans visée commerciale, mais pas d'un logiciel libre, dans la mesure où il était la propriété de l'éditeur d'Andrew Tanenbaum, selon les modalités classiques de la loi sur le *copyright*.

2. Linus TORVALDS, « What would you like to see most in minix ? », message posté sur le *newsgroup* comp.os.minix le 25 août 1991, en ligne : http://fr.wikisource.org/wiki/Naissance_de_Linux.

largement enlisé, et il ne présentait pas de réelle perspective d'achèvement à la sortie de la première version de Linux. Cette situation força quelque peu le mariage entre GNU et Linux.

L'union laissa à Richard Stallman un goût assez amer¹. Le créateur du logiciel libre n'avait eu aucun contact avec Linus Torvalds et les développeurs de Linux jusqu'en 1993, et la nouvelle génération lui témoignait peu de reconnaissance, eu égard au rôle fondateur qui avait été le sien. Ils nommaient le nouveau système Linux et non GNU/Linux, ce qui occultait le travail réalisé depuis 1984. Par ailleurs, de nouvelles distributions² indépendantes du projet GNU et de la *Free Software Foundation* se créaient : Debian ou Slackware par exemple. Celles-ci n'hésitaient pas à proposer à leurs utilisateurs des logiciels propriétaires en plus des logiciels libres qui constituaient le cœur du système, pratique en contradiction avec les principes intransigeants de Richard Stallman.

Du point de vue des objectifs et des valeurs défendus, l'attelage formé par GNU et Linux était ainsi relativement improbable. Il amalgamait un projet construit sur la base d'un message social clair et revendiqué (GNU), et un logiciel (Linux) écrit par des passionnés par simple goût de la programmation et amour de l'informatique. Richard Stallman avait démarré le projet GNU pour faire perdurer la culture *hacker* du MIT, et avait d'emblée conçu l'outil technique comme un « moyen pour atteindre un but social »³. Linus Torvalds ne s'était lancé dans la programmation de Linux que pour s'amuser⁴. Par ailleurs, quand l'un considérait l'existence de logiciels propriétaires comme un scandale éthique, l'autre n'hésitait pas à en utiliser lorsque cela s'avérait plus commode⁵. À l'évidence, la program-

1. Sam Williams, le biographe de Richard Stallman, use d'une comparaison assez parlante pour décrire les sentiments de celui-ci : « Tel Winston Churchill observant les troupes soviétiques déferler sur Berlin, Stallman éprouvait un sentiment mitigé bien compréhensible à l'heure de célébrer la « victoire » de GNU/Linux ». Sam WILLIAMS, Richard STALLMAN et Christophe MASUTTI, *Richard Stallman et la révolution du logiciel libre. Une biographie autorisée*, Paris : Eyrolles, 2010, p. 205.

2. Une distribution Linux est un ensemble cohérent de logiciels organisé autour du système GNU/Linux, qui peut inclure des logiciels non libres fonctionnant avec le système.

3. Richard M. STALLMAN, « Lecture at KTH », dans : Stockholm : Royal Institute of Technology, 1986, URL : <http://www.gnu.org/philosophy/stallman-kth.en.html>.

4. Linus TORVALDS et David DIAMOND, *Just for Fun : The Story of an Accidental Revolutionary*, New York : Harper Collins, 2001.

5. Lors d'une intervention à une conférence de *hackers* en 1996, Linus Torvalds choqua les « puristes » du logiciel libre en admettant être un utilisateur enthousiaste du logiciel *Power Point* développé par Microsoft.

mation de logiciels libres n'avait pas la même signification pour les deux hommes et leurs partisans respectifs. Ces différences laissaient présager quelques difficultés ultérieures.

2. La naissance du mouvement *open source*

Au milieu des années 1990, le monde des affaires commença, quoique d'abord assez timidement, à s'intéresser au logiciel libre. La société Red Hat fut créée en 1995 avec le projet d'éditer sa propre distribution de GNU/Linux, et de faire payer à ses clients tous les services afférents : formation, maintenance, personnalisation, etc. Elle suivait en cela les traces d'une entreprise pionnière, Cygnus Solutions, co-fondée en 1990 par Michael Tiemann. Celui-ci fut l'un des premiers à déceler dans le logiciel libre un potentiel commercial. Il décrit ainsi sa réception du « manifeste GNU » : « Cela ressemblait à de la polémique socialiste¹, mais j'y ai vu quelque chose de différent. J'y ai vu un *business plan* caché. »²

À mesure que les logiciels libres gagnaient en qualité et convertissaient de nouveaux utilisateurs, quelques entrepreneurs lui emboîtèrent le pas. En janvier 1998, les dirigeants de Netscape décidèrent d'ouvrir le code source de leur navigateur Web. Netscape était à l'époque un acteur majeur d'Internet, mais *Netscape Navigator* – qui dominait pourtant outrageusement le marché au milieu des années 1990 – perdait du terrain face à *Internet Explorer* de Microsoft. Rapidement suivie par la création de l'organisation Mozilla, la décision d'ouvrir le code source représentait un signal fort. Du point de vue technologique, elle témoignait de la confiance accordée aux méthodes de développement ouvertes et collaboratives. Du point de vue commercial, elle manifestait la conviction qu'il était possible de trouver de nouveaux modèles économiques tirant parti du logiciel

1. On notera ici l'emploi pour le moins relâché du terme « socialiste », qui tend souvent dans le contexte idéologique américain à englober toute forme de pensée exhibant une quelconque distance avec l'idéal du marché libre. On précisera aussi que Richard Stallman, bien qu'il ne fasse pas mystère de ses convictions politiques ancrées à gauche, n'a jamais présenté le mouvement du *free software* comme étant en lien avec une adhésion au socialisme comme idéologie. Bien au contraire, son insistance sur la dimension « éthique » du logiciel libre est précisément une manière d'essayer de dépasser les clivages politiques et partisans.

2. Michael TIEMANN, cité dans WILLIAMS, STALLMAN et MASUTTI, *op. cit.*, p. 181.

libre, afin de dépasser tant l'idéalisme souvent associé au *free software* que l'opportunisme mercantile du logiciel propriétaire¹.

En ce début d'année 1998, de nombreuses entreprises informatiques demeureraient malgré tout réticentes à s'engager dans le logiciel libre. Elles étaient notamment rebutées par l'appellation *free software*, spontanément associée à une idée de gratuité peu favorable aux affaires. C'est pour tenter de mettre fin à ces ambiguïtés et favoriser la pénétration du logiciel libre dans le monde de l'entreprise, que le terme *open source* fut forgé. Son usage se répandit comme une traînée de poudre, à un moment où l'enthousiasme autour d'Internet était à son comble dans les médias et chez les investisseurs. Richard Stallman ne tarda pas à désapprouver cette évolution terminologique. L'expression « *open source* » avait certes l'avantage d'éliminer la confusion entre liberté et gratuité, qu'il avait lui-même sans cesse cherché à désamorcer. Mais elle conduisait aussi à passer sous silence la question « éthique » de la liberté, qui avait toujours été au cœur de son combat. Autrement dit, elle faisait de la liberté des logiciels un simple moyen pour produire des technologies plus efficaces, et non une fin investie d'une haute valeur morale.

Dès « Le manifeste GNU », Richard Stallman avait en effet revendiqué son adhésion à une certaine « éthique kantienne », retenant de celle-ci le critère d'universalisation², adapté par ses soins aux enjeux spécifiques du logiciel libre moyennant un zeste de conséquentialisme³ : « Comme je n'aime pas les conséquences qu'entraînerait le fait que tout le monde

1. Pour de nombreux analystes, le bien-fondé de la démarche était cependant sensiblement plus sujet à caution en matière de stratégie d'entreprise que d'ingénierie logicielle. Dans les faits, ce fut un échec aussi bien technologique qu'économique, du moins dans un premier temps. L'ouverture du code source ne parvint pas à enrayer le déclin de l'entreprise, et le développement du navigateur sur la base du code libéré fut assez rapidement arrêté. Toutefois, l'initiative de Netscape accoucha indirectement d'une grande réussite, lorsque quelques années plus tard le navigateur *Firefox*, issu d'un sous-projet de *Mozilla*, commença à être loué pour ses performances, et à grignoter les parts de marché de Microsoft. Celui-ci réussit ainsi là où *Netscape Navigator* avait échoué.

2. Emmanuel KANT, *Critique de la raison pratique*, Paris : Presses Universitaires de France, 1993, p. 30. « Agis de telle sorte que la maxime de ta volonté puisse toujours valoir en même temps comme principe d'une législation universelle ».

3. La philosophie morale a pour habitude de nommer « éthique déontologique » toute théorie de type kantien, faisant dépendre la moralité d'un acte de sa conformité à un principe général et formel, indépendamment des conséquences ou des conditions particulières de l'action. Une telle approche s'oppose à l'éthique conséquentialiste, qui subordonne la moralité d'un acte à ses conséquences concrètes, par exemple au fait qu'il contribue à maximiser un intérêt général préalablement défini. La position éthique de Richard Stallman peut être

retienne l'information, je dois considérer qu'il est immoral pour chaque individu d'agir ainsi. »¹

Un tel rigorisme moral était totalement étranger aux promoteurs de l'approche *open source*, qui se livrèrent à une réorientation consciente et assumée de la signification du mouvement du logiciel libre. Leurs motivations combinaient une volonté pragmatique de ne pas rater le train de la croissance du secteur des nouvelles technologies avec une résistance viscérale au discours social tenu par Richard Stallman, perçu comme dangereux et idéologique. Le terme *open source* fut ainsi créé pour « se débarrasser de l'attitude moralisatrice et belliqueuse qui avait été associée au logiciel libre par le passé, et en promouvoir l'idée uniquement sur une base pragmatique et par un raisonnement économique, à l'image de ce qui avait motivé Netscape »². Les terminologies *free software* et *open source software* impliquaient par conséquent davantage qu'un débat sémantique. Elles mettaient en mots la différence d'approche, manifeste depuis quelques années déjà, entre le projet GNU et Linux. Elles séparaient le monde du logiciel libre entre un mouvement social porteur d'un discours éthique et une tendance managériale véhiculant un discours technique.

En 1998 ce clivage s'institutionnalisa, avec la création par Eric S. Raymond et Bruce Perens de l'*Open Source Initiative* (OSI)³. Cette nouvelle organisation se présentait à bien des égards comme la rivale de la *Free Software Foundation*. L'OSI commença ainsi à délivrer le label *OSI approved* aux logiciels dont les licences satisfaisaient aux critères de la définition de l'*open source*, moins restrictive et injonctive que celle du *free software*. Les divergences portaient notamment sur le principe du *copy-left*. Quand la *Free Software Foundation* défendait celui-ci, c'est-à-dire

présentée comme une sorte d'hybride assez étrange entre les deux, dans la mesure où elle reprend à la fois le critère d'universalisation typiquement déontologique, et le raisonnement téléologique propre au conséquentialisme.

1. Richard M. STALLMAN, « The GNU Manifesto ». URL : <http://www.gnu.org/gnu/manifesto.html>.

2. Michael TIEMANN, *History of the OSI*, 2012, URL : <http://www.opensource.org/history>.

3. Le dernier nommé se retira de l'organisation un an après sa création, regrettant l'opposition de l'OSI à la FSF et écrivant par la suite un texte pour expliquer pourquoi les principes mis en avant par la *Free Software Foundation* assuraient plus de libertés aux utilisateurs que ceux défendus par l'*Open Source Initiative*.

l'obligation d'offrir les mêmes libertés aux utilisateurs sur toutes les versions dérivées d'un logiciel libre, l'*Open Source Initiative* se contentait de permettre que les logiciels dérivés soient soumis aux mêmes conditions. Il s'agissait là d'une nuance importante, tant le principe du *copyleft* avait jusqu'alors représenté un élément majeur de l'identité du mouvement du logiciel libre.

Le développement de l'approche *open source* fut concomitant à la croissance fulgurante des entreprises de nouvelles technologies, et au gonflement de la bulle Internet. Les entreprises liées à l'*open source* profitèrent largement de cette euphorie boursière. La société *Red Hat* fit son entrée au *Nasdaq* en 1999, et, en décembre de la même année, *VA Linux* vit sa cotation monter en flèche, l'action gagnant 700% en une journée, le 10 décembre. L'allégresse des investisseurs provoqua une certaine excitation médiatique, à moins que ce ne fût l'inverse. Le logiciel libre devint, principalement sous l'appellation *open source*, un sujet de choix pour les médias généralistes américains à la fin de la décennie. En couverture des *news magazines*, il n'était plus rare de voir des photos de Linus Torvalds. Le succès médiatique du mouvement *open source* contribua à modifier l'image du logiciel libre. Celui-ci se fit connaître des milieux d'affaire et – un peu – du grand public, mais moyennant une dilution de son message social et éthique intransigeant. Selon Richard Stallman, l'*open source* ne fournissait en effet qu'une « alternative édulcorée »¹ au discours développé depuis l'origine par le *free software*, réduisant le logiciel libre à n'être qu'un modèle de développement efficace.

Ce déplacement fut largement impulsé par les écrits de l'informaticien et anthropologue autodidacte Eric S. Raymond, fervent partisan de l'approche *open source*. Dans son texte le plus célèbre, « La cathédrale et le bazar », Raymond compare deux styles de développement : le modèle « cathédrale » et le modèle « bazar ». Le premier est, selon lui, caractéristique de la majorité des logiciels commerciaux, mais aussi dans une certaine mesure du projet GNU. Il repose sur un petit groupe de gens « travaillant à l'écart du monde »², au sein d'une organisation stricte et centralisée. Le second a été « inventé » par Linus Torvalds et la communauté

1. Richard M. STALLMAN, « Thus Spake Stallman », dans : *Interview* (2000), URL : <http://slashdot.org/story/00/05/01/1052216/thus-spake-stallman>.

2. Eric S. RAYMOND, *The Cathedral and the Bazaar*, Sebastopol, CA : O'Reilly, 2001, URL : <http://www.catb.org/esr/writings/homesteading/>. Traduit de l'américain

des développeurs du noyau Linux. Eric Raymond le décrit comme ouvert à tous et « grouillant de rituels et d'approches différentes ». Sa fécondité tiendrait à l'investissement personnel des programmeurs, aux mécanismes de réputation et d'autorégulation, et aux mises à jour fréquentes de nouvelles versions, boguées mais rapidement corrigées par la communauté. De cette dernière caractéristique, Eric Raymond tire ce qu'il nomme la « loi de Linus » : « Étant donnés suffisamment d'observateurs, tous les bogues sautent aux yeux ». Il en conclut que le génie de Linus Torvalds ne tient pas à « la construction du noyau de Linux en lui-même, mais plutôt à son invention du modèle de développement de Linux », plus efficace que celui de n'importe quel « projet fermé ».

Dans le contexte de l'époque, le texte avait une dimension polémique indéniable, Richard Stallman et le projet GNU y étant implicitement accusés de n'avoir pas su ouvrir suffisamment leurs pratiques de développement. Ce reproche pouvait néanmoins apparaître assez injuste, tout comme il pouvait sembler légèrement exagéré de présenter Linus Torvalds comme « l'inventeur » d'un nouveau modèle de production logicielle. En effet, le développement de certaines versions d'Unix, ou celui d'Emacs tel qu'il fut mené au MIT, mettaient déjà en œuvre des processus de collaboration du même type. Hal Abelson, étudiant au MIT dans les années 1970 et membre fondateur de la *Free Software Foundation*, décrit par exemple le projet Emacs de la façon suivante : « Sa structure était si robuste que des gens du monde entier pouvaient y collaborer ou y contribuer sans concertation préalable. Je ne sais pas si cela avait jamais été réalisé auparavant »¹. Une telle description est assurément assez proche du modèle du « bazar » théorisé par Eric Raymond. Il appartient cependant au développement de Linux d'avoir mis en application ces principes à une échelle jusqu'alors inconnue, ce qui fut permis par l'expansion d'Internet.

La prose d'Eric Raymond eut également pour effet d'étendre l'intérêt et l'admiration pour le modèle de production du logiciel libre hors du cercle des *hackers*. En 1998, dans des notes qui étaient censées demeurer confidentielles (les *Halloween Documents*), un responsable de Microsoft reconnaissait que la méthodologie de développement *open source* produi-

par Sébastien Blondeel, 11 août 1998, en ligne : <http://www.linux-france.org/article/these/cathedrale-bazar/>.

1. WILLIAMS, STALLMAN et MASUTTI, *op. cit.*, p. 117.

sait des logiciels aussi – sinon plus – robustes que les alternatives commerciales. Plus globalement, de nombreux discours sur les vertus quasi magiques de l’auto-organisation fleurirent à cette époque, érigeant le logiciel libre en paragon d’une « intelligence collective » propulsée par Internet. La métaphore du « bazar » laissait en effet supposer que les communautés *open source* fonctionnaient de manière anarchique et étaient dénuées de structure d’autorité. Bien qu’elle fut largement mythique¹, une telle vision ne pouvait assurément manquer d’impressionner les esprits.

Le clivage entre *free software* et *open source* était donc très net en cette fin des années 1990. D’un côté, il existait un « mouvement social », c’est-à-dire un acteur collectif (certes très lié à la figure charismatique de Richard M. Stallman) défendant à travers la question du logiciel des valeurs de portée générale : circulation de l’information, partage de la connaissance, autonomie par rapport aux technologies. De l’autre, il y avait une « méthodologie de développement », tirant parti des nouvelles possibilités d’interconnexion à grande échelle fournies par Internet et efficacement promue par quelques zéloteurs habiles. Essayons de préciser les spécificités de chacune de ces deux postures.

3. Le pragmatisme contre l’idéologie

L’une des caractéristiques les plus frappantes du discours de l’*open source* fut sans nul doute son extrême insistance à se présenter comme dépourvu de toute idéologie, ce terme faisant en quelque sorte figure de repoussoir ultime. Les propos de Linus Torvalds en la matière se sont rarement embarrassés de subtilités. Dans un message posté sur la liste de diffusion du noyau Linux, il affirmait très franchement : « Je pense que l’idéologie, c’est nul »². Il précisait ensuite que

[...] ce monde pourrait être un endroit bien meilleur si les gens avaient moins d’idéologie, et plus de « je le fais parce que c’est

1. Nikolai BEZROUKOV, « A Second Look at the Cathedral and the Bazar », dans : *First Monday* 4.12 (1999), URL : <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/708/618>.

2. En VO : « *I think ideology sucks* » (Linus TORVALDS, « Re : [PATCH] Remove Bitkeeper Documentation from Linux Tree », message posté le 20 avril 2002 sur la liste de diffusion du noyau Linux, en ligne : <http://lkml.indiana.edu/hypermail/linux/kernel/0204.2/1018.html>).

FUN et parce que d'autres trouveront peut-être ça utile, non parce que j'ai une religion. »¹

Ce refus de l'idéologie a toujours eu pour corollaire une revendication de pragmatisme. Dans un billet au titre provocateur, « Free Software is Dead. Long Live Open Source », le chroniqueur Matt Asay a exemplairement résumé cette posture intellectuelle :

Par « pragmatisme », je ne veux pas dire « capitulation » au sens où l'*open source* en viendrait à ressembler au monde propriétaire qu'il cherche à remplacer. Plus précisément, je sous-entends que plus l'*open source* s'adresse à un large public, plus il apprend à faire des compromis ; compromis qui le rendent plus fort, et non plus faible.²

Le couple idéologie/pragmatisme en dit beaucoup sur les conceptions au cœur de l'approche *open source*. Ce qui est mis en avant est la légitimité et l'efficacité dans le domaine informatique de ce qu'on pourrait appeler une « politique des petits pas », ou encore une « politique de l'entendement »³. Il faut entendre par là que l'affirmation d'une posture de concurrence avec le logiciel propriétaire n'est pas vue comme excluant la recherche de compromis ou d'arrangements, qui ne sont pas considérés comme des compromissions. Le fait que les distributions commerciales de GNU/Linux (Red Hat, Ubuntu, SuSe, etc.) incluent des logiciels non libres en est un bon exemple. L'approche *open source* défend cette entorse à l'esprit originel du *free software*, avec des arguments de commodité pour l'utilisateur, de performance, ou de stratégie commerciale. En effet, le mouvement *open source* a toujours associé son esprit de conciliation à une exigence d'efficacité. Il n'a cessé de revendiquer ses résultats concrets, à l'inverse des effets (censément) paralysants des principes intransigeants défendus par

1. *idem*.

2. Matt ASAY, « Free software is dead. Long live open source », dans : *CNET* (2009), URL : <http://news.cnet.com/8301-135053-10361785-16.html>.

3. La notion de « politique de l'entendement », et la notion symétrique de « politique de la raison », ont été développées par le philosophe Alain, et discutées notamment par Raymond Aron et Maurice Merleau-Ponty. Dans *Les aventures de la dialectique*, ce dernier parlait de la politique de l'entendement comme de celle qui « prend l'homme comme il est, à l'œuvre dans un monde obscur, résout les problèmes un à un, cherche chaque fois à faire passer dans les choses un peu des valeurs que l'homme, quand il est seul, discerne sans hésitation, et ne connaît d'autre stratégie que la somme de ces actions de harcèlement ». Maurice MERLEAU-PONTY, *Les aventures de la dialectique*, Paris : Gallimard, 1955, p. 10.

les adeptes du *free software*. Les grandes déclarations de ces derniers sont ainsi présentées comme à la fois extrémistes et improductives, du fait de leur refus de tout compromis et de la manière dont elles culpabilisent les utilisateurs de logiciels propriétaires. Linus Torvalds avoue par exemple trouver « les personnes qui ont de grandes visions », certes « très intéressantes mais souvent un peu effrayantes ». Il conseille par ailleurs aux développeurs « de ne pas refaire le monde, mais d'essayer d'apporter de petites améliorations précises »¹.

Il est vrai que les partisans de l'*open source* ont toujours fait de la qualité des logiciels leur principal objectif. Linus Torvalds évoque les raisons qui poussent à maintenir le code source ouvert comme suit : « Ce n'est pas pour *partager l'information* en soi : partager l'information fait partie des outils pour créer de meilleurs logiciels »². Un discours d'ingénieur assez classique remplace ainsi l'affirmation d'un principe valant de manière absolue : le partage de l'information n'est plus qu'un moyen et non une fin en soi. L'excellence technologique est en outre appréhendée comme le levier permettant de convaincre les entreprises et les particuliers de venir au logiciel libre. Pour les partisans du pragmatisme, il est inutile de stigmatiser ou de condamner moralement le logiciel propriétaire. Il y a simplement à constater que celui-ci risque rapidement d'être dépassé par ses concurrents *open source* du strict point de vue de la qualité :

Il est probable qu'à terme, la culture du logiciel dont le code source est ouvert triomphera, non pas parce qu'il est moralement bon de coopérer, non pas parce qu'il est moralement mal de « clôturer » le logiciel (en supposant que vous soyez d'accord avec la deuxième assertion ; ni Linus ni moi ne le sommes), mais simplement parce que le monde dont le code source est fermé ne peut pas gagner une course aux armements évolutive contre des communautés de logiciel libre, qui peuvent mettre sur le problème un temps humain cumulé plus important de plusieurs ordres de grandeurs.³

1. LINUS TORVALDS, « Le concept de l'*open source* oblige distributeurs et développeurs à rester honnêtes », entretien avec Stephen Shankland, *CNET News.com*, 3 janvier 2005, article repris et traduit par *zdnnet.fr*, en ligne sur <http://www.zdnnet.fr>.

2. LINUS TORVALDS, cité par Bruce BYFIELD, « Linus explains why open source works », dans : *Linux.com* (2007), URL : <http://archive09.linux.com/feature/118380>.

3. RAYMOND, *op. cit.*

Cette position est diamétralement opposée à celle de Richard Stallman, qui n'a cessé de clamer que la performance technologique était pour lui secondaire par rapport aux objectifs sociaux du *free software* :

Pour moi, le logiciel libre est avant tout une question de liberté et de communauté. Nous avons besoin du logiciel libre pour que les utilisateurs d'ordinateurs soient libres de coopérer. C'est pour cette seule raison que j'ai décidé de rejeter le logiciel non libre. Que le logiciel libre aboutisse aussi à du logiciel efficace et puissant a été une surprise pour moi, et je m'en réjouis. Mais c'est un bonus. J'aurais choisi le logiciel libre, même s'il avait été moins efficace et moins puissant – parce que je ne brade pas ma liberté pour de simples questions de convenances.¹

Le mouvement *open source* a ainsi largement réussi à adapter le logiciel libre aux impératifs du marché de l'informatique, en tenant aux décideurs un discours répondant à leurs préoccupations : mutualisation des coûts, amélioration de la qualité, élimination du *vendor lock-in*, etc. Bien que le raccourci soit tentant, l'antagonisme *free software/open source* est toutefois un peu plus complexe qu'une opposition entre une approche *pro-business* et une approche *anti-business*. Richard Stallman a par exemple toujours affirmé qu'il ne voyait pas d'incompatibilité de principe entre le logiciel libre et les entreprises². De même, il n'a jamais été opposé au fait que les programmeurs soient payés pour leur travail³. En revanche, il a toujours été inflexible sur le respect et la pérennité des quatre libertés

1. Richard M. STALLMAN, « La passion du libre : entretien avec Jérôme Gleizes et Aris Papatheodorou », dans : *Multitudes* 1 (2000), URL : <http://multitudes.samizdat.net/La-passion-du-libre>.

2. Dans un texte écrit en 1998 pour présenter le projet GNU, il précise que « la philosophie du logiciel libre rejette une pratique commerciale spécifique très répandue dans l'industrie du logiciel », mais qu'elle n'est pas « hostile au monde des affaires ». Pour être sûr d'être bien compris, il ajoute que « quand des entreprises respectent la liberté des utilisateurs », il leur souhaite tout simplement « de réussir ». Richard M. STALLMAN, « The Gnu Project », dans : *Free Software, Free Society : Selected Essays of Richard M. Stallman*, Boston : GNU Press, 2002, p. 17–32.

3. Dans le « Manifeste GNU », il affirme « qu'il n'y a rien de mal à vouloir être payé pour son travail, ou à chercher à augmenter ses revenus, tant que l'on n'utilise pas de méthodes destructives ». Il précise ensuite qu'il existe de « nombreuses façons pour les programmeurs de gagner leur vie sans avoir à vendre leur droit d'utiliser le programme », même si celles-ci sont sans doute légèrement moins rémunératrices que les pratiques en vigueur dans le monde propriétaire. Il ajoute enfin que « puisqu'il n'est pas considéré comme injuste que les caissières gagnent ce qu'elles gagnent », il n'y aurait pas davantage d'injustice à ce que les programmeurs aient un revenu équivalent, sachant qu'ils gagneraient sans doute

du logiciel libre, celles-ci devant selon lui primer sur toute autre considération d'ordre technique ou économique. Richard Stallman affirme ainsi qu'il n'est jamais légitime de renoncer à tout ou partie de ces libertés, et qu'aucun compromis sur ce point n'est éthiquement acceptable, quels qu'en soient les bénéfices.

L'un des principaux points de rupture avec l'approche de l'*open source* tient finalement à cette intransigeance, et à la volonté systématique de placer le débat sur le terrain moral. À l'inverse, le rejet de « l'idéologie » manifesté par le mouvement *open source* doit être compris d'une part comme une marque de désintérêt pour tout ce qui ne relève pas de la volonté de produire les meilleurs logiciels possibles, d'autre part comme une revendication de flexibilité dans les principes. Il s'agit d'une prise de distance vis-à-vis de tout ce qui pourrait nuire à l'adaptation au monde tel qu'il est, et ainsi retarder les succès techniques et économiques du logiciel libre. « L'idéologie », ce sont donc tous les discours sociaux et éthiques qui marquent une volonté de « changer le monde » à travers la programmation informatique libre, et impliquent qu'il s'agit là d'un projet dont les finalités excèdent la production de technologies plus pratiques et plus efficaces. Le « pragmatisme » revendiqué témoigne d'une inclination à mettre au centre du mouvement du logiciel libre les pratiques de programmation collaborative, et non les argumentations en sa faveur. Il est surtout synonyme de modération et de réalisme, autrement dit d'une acceptation des règles du jeu économique et social.

On remarquera pour finir que dans ses attaques contre le *free software*, le mouvement *open source* a reconduit une stratégie argumentative bien connue, consistant à faire de l'autre un idéologue tout en se proclamant soi-même totalement dépourvu d'idéologie. Le marxisme auto-proclamé « scientifique » en a fait grand usage en d'autres temps, et – à l'autre extrémité du spectre politique – le discours sur la « fin des idéologies » en a fourni pléthore d'exemples dans le contexte de la guerre froide. En 1958, Murray Dyer affirmait ainsi :

Nous ne pouvons opposer à l'idéologie une autre idéologie pour une raison simple : nous n'avons pas d'idéologie. Les Soviétiques, eux,

plus d'argent tout de même. Richard M. STALLMAN, « The Gnu Manifesto », dans : *Free Software, Free Society : Selected Essays of Richard M. Stallman*, Boston : GNU Press, 2002, p. 33–42.

ont largement démontré qu'ils voulaient imposer à l'humanité leur doctrine et leur idéologie.¹

Toutes choses égales par ailleurs, remplacez « les Soviétiques » par « les partisans de Richard Stallman » et vous aurez une synthèse convaincante des attaques portées par l'*open source* contre le *free software* !

4. L'idéologue n'est pas nécessairement celui qu'on croit

Bien évidemment, une telle pureté « non-idéologique » paraît quelque peu suspecte à quiconque fait preuve d'un tant soit peu d'esprit critique. Il est clair en effet que, malgré ce que les promoteurs de l'*open source* aiment à raconter, les positions qu'ils défendent impliquent une opinion tranchée quant aux buts du logiciel libre et aux moyens appropriés pour les atteindre. Il y a bien un discours de l'*open source*, quand bien même ce discours consiste à dire que seules les pratiques importent et qu'il faut être « pragmatique » avant tout. En ce sens, le *free software* n'est ni plus ni moins « idéologique » que l'*open source*. Dans les deux cas, nous avons affaire à un ensemble de représentations, d'idées et de jugements concernant la place du mouvement du logiciel libre dans le monde contemporain et les principes qui doivent le guider.

Les divergences entre ces deux visions concurrentes sont évidentes. Elles portent sur la définition du mouvement du logiciel libre, sur ses finalités, sur son rapport à la performance technique et sur sa relation aux entreprises du secteur de l'informatique. D'un côté, le *free software* se pense comme un « mouvement social » et adopte un positionnement éthique intransigeant, le conduisant à considérer l'excellence technologique et la rentabilité économique comme des questions secondaires. De l'autre, l'*open source* défend une approche technico-économique, plus flexible sur les principes, et dont le « pragmatisme » affiché n'est pas une absence de parti pris mais une mise à distance des finalités sociales poursuivies par le *free software*.

Il vaut donc prendre la peine de réinterroger le rejet de « l'idéologie » affirmé par les tenants de l'*open source*. À l'évidence, il s'agit moins d'un

1. Cité par Armand MATTELART, *Histoire de l'utopie planétaire : de la cité prophétique à la société globale*, Paris : la Découverte, 2009, p. 305.

refus de voir le mouvement du logiciel libre tenir un certain discours sur lui-même, que d'une critique portant sur le contenu particulier du discours du *free software*. Cette critique insiste sur les aspects moralisateurs, idéalistes et subversifs du propos militant développé par Richard Stallman et ses partisans. Autrement dit, ce qui apparaît comme « idéologique » aux défenseurs de l'*open source* est fondamentalement l'écart avec l'existant instauré par le discours du *free software* : écart entre la pureté des principes moraux défendus et l'opportunisme de fait des comportements individuels ; écart entre l'abstraction des principes affirmés et les exigences concrètes de l'économie du logiciel ; écart entre l'idéal social esquissé et la réalité du monde contemporain. Cet écart est perçu comme une marque d'irréalisme et un facteur d'inefficacité. À l'inverse, le discours de l'*open source* est toujours apparu ajusté au contexte économique et social. Lorsqu'il a émergé à la fin des années 1990, il collait parfaitement à une période qui voyait la bourse s'enthousiasmer pour les entreprises liées à Internet, et les médias vanter l'innovation technologique comme moteur de la croissance et du progrès social.

Ces deux grandes attitudes – s'adapter au monde tel qu'il est ou essayer de le subvertir – peuvent être qualifiées par deux concepts : idéologie et utopie. Le concept d'idéologie désigne toutes les forces qui tendent à légitimer l'ordre social existant et à le perpétuer. L'utopie désigne au contraire tout ce qui cherche à subvertir l'existant et à ouvrir des brèches dans le monde social tel qu'il s'impose aux individus¹. Dans cette perspective, le mouvement *open source* est sans conteste plus « idéologique » que le *free software* : il s'est toujours efforcé de ne pas bousculer trop rudement les habitudes du plus grand nombre, et s'est glissé avec douceur dans les draps du « capitalisme cognitif »².

À l'inverse, le *free software* comporte une dimension utopique très nette : sa rigidité doctrinale l'a poussé à refuser tout compromis « réaliste » avec l'informatique propriétaire et à maintenir un élan subversif. Au risque de l'isolement et de l'enlèvement. On accordera donc aux tenants du « pragmatisme » que leur flexibilité en matière de principes a

1. Karl MANNHEIM, *Idéologie et utopie*, trad. par Jean-Luc EVARD, Paris : Éditions de la Maison des sciences de l'homme, 2006. Paul RICOEUR, *L'idéologie et l'utopie*, Paris : Éd. du Seuil, 1997.

2. Yann MOULIER BOUTANG, *Le capitalisme cognitif : la nouvelle grande transformation*, Multitudes-idées, Paris : Éd. Amsterdam, 2007.

davantage contribué à faire progresser les parts de marché du logiciel libre que l'intransigeance de Richard Stallman. On ne pourra nier en revanche, que leur obsession à se présenter comme « non-idéologiques » a toujours constitué au mieux une posture habile, au pire une contre-vérité de la pire espèce.

5. Un débat obsolète ?

Tout compte fait, faut-il vraiment accorder de l'importance à ce débat entre *free software* et *open source* ? Après tout, les divergences « philosophiques » n'empêchent ni les collaborations, ni l'existence d'une seule et même catégorie de logiciels : les *Free and Open Source Software*, ou *FOSS*. Néanmoins, du point de vue institutionnel, le clivage reste important. Aux États-Unis, la *Free Software Foundation* (FSF) est toujours concurrencée par l'*Open Source Initiative* (OSI), bien que cette dernière organisation paraisse en déclin depuis quelques années. En France, il existe deux grandes associations de soutien au logiciel libre : l'April (Association pour la promotion et la recherche en informatique libre) et l'AFUL (Association Francophone des Utilisateurs de logiciels Libres). L'April, la première à avoir vu le jour en 1996, soutient sans réserve les positions de la *Free Software Foundation* et affirme défendre le logiciel libre pour une raison éthique. L'AFUL, créée en 1998, revendique quant à elle son appartenance au mouvement *open source* et affiche sa spécificité par rapport à son aînée : « Alors que l'April préfère se placer en premier sur le plan des principes, l'AFUL met en avant les avantages concrets des logiciels et ressources libres »¹. Enfin, les deux camps ont chacun leur grande manifestation annuelle. Les *Rencontres Mondiales du Logiciel Libre* (RMLL) s'affichent comme un rendez-vous non commercial et festif, alors que l'*Open World Forum* accueille chaque année dans une ambiance plus policée les principales entreprises de l'*open source* et leurs représentants les plus prestigieux.

Par ailleurs, la controverse entre *free software* et *open source* reflète assez souvent un vrai désaccord politique sous-jacent, quand bien même les acteurs du débat ne présentent presque jamais les choses sous cet angle.

1. AFUL, « Quelles différences entre l'AFUL et l'April ? », en ligne : <http://aful.org/association/differences-aful-april>, version du 01/04/2012.

Les partisans les plus en vue de l'*open source* expriment ainsi une grande confiance dans les vertus du marché et une méfiance prononcée vis-à-vis de l'État. Pour Linus Torvalds « si l'*open source* a une influence, c'est bien celle de rendre possible le capitalisme dans le secteur du logiciel », et de promouvoir ainsi le « marché libre »¹. Eric Raymond n'a lui jamais fait mystère de ses convictions politiques libertariennes. Il défend la réduction drastique de la sphère d'action du gouvernement et une extension maximale des libertés individuelles, jusqu'au droit pour chacun de détenir une arme à feu. C'est peu dire que Richard Stallman manifeste beaucoup plus de méfiance envers le pouvoir des grandes entreprises. Il s'affirme ainsi partisan de « réguler le monde des affaires de toutes les façons nécessaires pour protéger le bien-être public et la démocratie »², et va désormais jusqu'à affirmer que la lutte du *free software* « fait partie d'une lutte plus large qui est la lutte contre l'empire des entreprises »³. À un mouvement clairement inscrit dans la grande tradition du libéralisme économique et politique s'oppose ainsi un mouvement social critique, dont le militantisme n'est pas directement politique, mais implique néanmoins une mise en cause *de fait* du capitalisme néolibéral et des « nouvelles enclosures »⁴ que celui-ci installe.

Quand bien même on n'en ferait pas une lecture explicitement politique, *open source* et *free software* font apparaître les tentations opposées auxquelles le mouvement du logiciel libre se trouve confronté. D'un côté, la stratégie pragmatique passant par la conquête des entreprises, le gain de parts de marché, et l'abandon de toute velléité de transformation sociale radicale ne manque pas d'attraits. Son efficacité a été en grande partie démontrée, et son caractère modéré a de quoi séduire ceux que les grandes envies de changement inquiètent. De l'autre côté, la stratégie utopique semble plus exigeante et plus incertaine. Elle est néanmoins plus fidèle à

1. Linus TORVALDS, cité par Don TAPSCOTT et Anthony D. WILLIAMS, *Wikinomics. Wikipédia, Linux, YouTube... : comment l'intelligence collaborative bouleverse l'économie*, trad. par Brigitte VADÉ, Paris, France : Pearson Village Mondial, 2006, p. 109.

2. Richard M. STALLMAN, « RMS AMA. Richard Stallman answers your top 25 questions », dans : *Reddit.com* (2010), URL : <http://blog.reddit.com/2010/07/rms-ama.html>.

3. Richard M. STALLMAN, « Avec Hadopi, la France n'est pas un pays libre. Entretien avec Martin Untersinger », dans : *Rue 89* (2011), URL : <http://www.rue89.com/2011/10/29/richard-stallman-avec-hadopi-la-france-nest-pas-un-pays-libre-225958>.

4. James BOYLE, « The Second Enclosure Movement and the Construction of the Public Domain », dans : *Law and Contemporary Problems* 66.1-2 (2003), p. 33-74, URL : <http://www.law.duke.edu/pd/papers/boyle.pdf>.

l'esprit originel du logiciel libre, plus ambitieuse, et peut-être plus à même de répondre à certaines menaces actuelles.

En effet, les problèmes que posent les développements récents de l'informatique et d'Internet ne sont pas moins aigus que ceux de la fin des années 1990, et le *free software* a assurément des choses à en dire. Dans un monde de plus en plus technicisé, le mantra de Richard Stallman selon lequel il faut pouvoir contrôler nos machines pour ne pas être contrôlés par elles n'est pas dénué de pertinence. Et il est assurément intéressant qu'un mouvement de « techniciens » en soit venu à dénoncer le fétichisme technologique (Richard Stallman parle de *priesthood of technology*) et l'aliénation représentée par le consumérisme *high-tech*, dont Apple est aujourd'hui le symbole éclatant. Promouvoir un autre rapport à la technologie par le déploiement de pratiques alternatives *et* une réflexion sur ce qui se joue dans notre rapport aux objets techniques, tel est l'un des grands objectifs du mouvement du *free software*. Objectif ambitieux sans doute, mais nullement chimérique, comme en témoigne l'extension des principes du logiciel libre à d'autres technologies, avec le succès d'outils comme Arduino et le développement plus général du *Do It Yourself* : hackerspaces, Fab Lab, garage biology, repair cafés, etc. Il y a bien là les ferments d'une utopie, prônant la réappropriation de la technique, la recréation de sociabilités locales et l'investissement dans des activités non-marchandes intrinsèquement gratifiantes. L'avenir dira si celle-ci n'était qu'un rêve caressé par quelques passionnés ou une véritable force de transformation de l'existant.

Références

- ASAY, Matt, « Free software is dead. Long live open source », dans : *CNET* (2009), URL : <http://news.cnet.com/8301-135053-10361785-16.html>.
- BEZROUKOV, Nikolai, « A Second Look at the Cathedral and the Bazar », dans : *First Monday* 4.12 (1999), URL : <http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/708/618>.
- BOYLE, James, « The Second Enclosure Movement and the Construction of the Public Domain », dans : *Law and Contemporary Problems* 66.1-2 (2003), p. 33-74, URL : <http://www.law.duke.edu/pd/papers/boyle.pdf>.

- BYFIELD, Bruce, « Linus explains why open source works », dans : *Linux.com* (2007), URL : <http://archive09.linux.com/feature/118380>.
- KANT, Emmanuel, *Critique de la raison pratique*, Paris : Presses Universitaires de France, 1993.
- MANNHEIM, Karl, *Idéologie et utopie*, trad. par Jean-Luc EVARD, Paris : Éditions de la Maison des sciences de l'homme, 2006.
- MATTELART, Armand, *Histoire de l'utopie planétaire : de la cité prophétique à la société globale*, Paris : la Découverte, 2009.
- MERLEAU-PONTY, Maurice, *Les aventures de la dialectique*, Paris : Gallimard, 1955.
- MOULIER BOUTANG, Yann, *Le capitalisme cognitif : la nouvelle grande transformation*, Multitudes-idées, Paris : Éd. Amsterdam, 2007.
- RAYMOND, Eric S., *The Cathedral and the Bazaar*, Sebastopol, CA : O'Reilly, 2001, URL : <http://www.catb.org/esr/writings/homesteading/>.
- RICOEUR, Paul, *L'idéologie et l'utopie*, Paris : Éd. du Seuil, 1997.
- STALLMAN, Richard M., « Avec Hadopi, la France n'est pas un pays libre. Entretien avec Martin Untersinger », dans : *Rue 89* (2011), URL : <http://www.rue89.com/2011/10/29/richard-stallman-avec-hadopi-la-france-nest-pas-un-pays-libre-225958>.
- « La passion du libre : entretien avec Jérôme Gleizes et Aris Papatheodorou », dans : *Multitudes* 1 (2000), URL : <http://multitudes.samizdat.net/La-passion-du-libre>.
 - « Lecture at KTH », dans : Stockholm : Royal Institute of Technology, 1986, URL : <http://www.gnu.org/philosophy/stallman-kth.en.html>.
 - « RMS AMA. Richard Stallman answers your top 25 questions », dans : *Reddit.com* (2010), URL : <http://blog.reddit.com/2010/07/rms-ama.html>.
 - « The Gnu Manifesto », dans : *Free Software, Free Society : Selected Essays of Richard M. Stallman*, Boston : GNU Press, 2002, p. 33–42.
 - « The Gnu Project », dans : *Free Software, Free Society : Selected Essays of Richard M. Stallman*, Boston : GNU Press, 2002, p. 17–32.
 - « Thus Spake Stallman », dans : *Interview* (2000), URL : <http://slashdot.org/story/00/05/01/1052216/thus-spake-stallman>.
- TAPSCOTT, Don et Anthony D. WILLIAMS, *Wikinomics. Wikipédia, Linux, YouTube... : comment l'intelligence collaborative bouleverse l'éco-*

nomie, trad. par Brigitte VADÉ, Paris, France : Pearson Village Mondial, 2006.

TIEMANN, Michael, *History of the OSI*, 2012, URL : <http://www.opensource.org/history>.

TORVALDS, Linus et David DIAMOND, *Just for Fun : The Story of an Accidental Revolutionary*, New York : Harper Collins, 2001.

WILLIAMS, Sam, Richard STALLMAN et Christophe MASUTTI, *Richard Stallman et la révolution du logiciel libre. Une biographie autorisée*, Paris : Eyrolles, 2010.

IV

Licences libres

L'évolution des licences libres et *open source* : critères, finalités et complétude ?

Benjamin JEAN

Au premier regard, l'intérêt d'une étude historique ayant pour objet les licences libres n'est pas évident et l'on tend généralement, avec raison, à considérer que la véritable valeur du Libre réside dans les projets qui le composent (GNU-Linux, Firefox, LibreOffice, Wikipédia, etc.). Ainsi, complexes et hétérogènes, les documents contractuels associés ne seraient qu'une formalisation nécessaire, une attache obligatoire du Libre au système légal conséquente à la juridicisation de la société actuelle.

Une telle approche nierait la place toute particulière des licences au sein du mouvement du Libre en cristallisant les tensions constitutives du système, assurant l'équilibre d'une équation aux multiples variables. Dès le départ, les licences libres se positionnent en rupture avec les usages traditionnels attirant un nombre important d'« amateurs » (au regard de l'importance qu'ils y attachent et de leur grande connaissance du sujet, la plupart étant par ailleurs non juristes), une image qui dénote avec l'austérité souvent associée aux documents contractuels et remet en cause le monopole du juriste (lui-même souvent dépassé par la complexité d'un document aussi juridique que social)... L'usage du droit s'éloigne donc de la technicité qui le caractérise aujourd'hui pour renouer avec sa mission première d'organisation des relations sociales : formalisant les règles

relatives aux rapports entre les contributeurs d'un projet et entre les communautés, assurant sécurité et une pérennité à leur collaboration. Bien qu'il soit récent, le système est solide et on ne peut que saluer le nombre réduit de contentieux (la vocation du droit étant aussi d'éviter les litiges) : moins d'une dizaine à l'échelle internationale, si on se concentre sur ceux ayant réellement une spécificité attachée au Libre et que l'on écarte les litiges potentiels très souvent réglés par le biais de transactions¹. Toujours dans ce rapport qu'entretiennent les licences libres et le droit, il est important de remarquer que les rédacteurs des licences jouissent généralement d'une position « à part », relativement privilégiée : législateurs d'une communauté réduite, ils disposent de grands pouvoirs et donc de grandes responsabilités.

Cette situation n'est cependant que la résultante d'une lente évolution. Même s'ils ne le furent formellement que quelques années plus tard, les logiciels libres sont conceptuellement nés avec l'informatique : dès la massification des ordinateurs, les utilisateurs ont commencé à partager leurs programmes et à contribuer sur ceux de leurs pairs (de fait, la plupart des systèmes d'exploitation étaient ainsi libres jusqu'aux années 1970). Les logiciels étaient alors dénués de toute valeur économique intrinsèque et développés collectivement par un nombre restreint de personnes issues de la Recherche, de la Défense et des quelques entreprises vendant du matériel informatique. Aucune propriété n'était, pour ainsi dire, revendiquée sur ces derniers. Néanmoins, la démocratisation de l'informatique a favorisé le développement d'un marché du logiciel qui a motivé la transformation de ces produits immatériels en produits de consommation, à l'instar des livres ou des disques, par la reconnaissance d'un droit de propriété « intellectuelle » (en l'occurrence le droit d'auteur). Le Copyright Office américain ouvrit l'enregistrement (nécessaire pour bénéficier du copyright) d'œuvres logicielles dès les années 1960, sans préjuger du bénéfice de la protection par le copyright alors très incertaine. Mais ce n'est qu'en 1976, au travers du Copyright Act, et plus spécifiquement en 1980 par le Computer Software Copyright Act, qu'une telle protection fut reconnue par la Loi². Ce fut alors l'âge d'or des éditeurs de logiciels (qui dé-

1. L'initiative *gpl-violations.org* évalue à plus de 150 le nombre de dossiers traités en dehors des tribunaux.

2. La situation fut relativement similaire en France avec une doctrine qui accepta progressivement l'idée d'une protection du logiciel au même titre qu'une œuvre littéraire et la

mocratisèrent le *shrink wrap*, contrat associé à des logiciels vendus « sur étagère », et accepté par l'ouverture du produit), sur lequel reposent toujours des sociétés telles que Microsoft, Oracle, SAP, etc. Parallèlement au développement de cette industrie (intimement liée à l'extension des droits qu'elle entretient par un lobbying intensif), certains logiciels continuèrent néanmoins à être créés collectivement toujours au sein de la recherche mais aussi parmi une communauté croissante de hackers. Ces derniers, n'ayant pas les moyens ou ne souhaitant pas acheter de telles licences (alors généralement négociées par et pour les seules universités¹), mirent en place des règles de non-appropriation (non exclusivité) au travers des licences libres, rétablissant une confiance nécessaire au développement d'un projet commun.

S'appuyant sur le droit d'auteur (et sur un nombre croissant de nouveaux droits exclusifs) qui joue un rôle important et structurant, le Libre s'est étendu dans le secteur du logiciel et au-delà, en introduisant lorsqu'il le fallait de nouvelles problématiques, spécifiques ou générales à ces droits. Plus précisément, le système des licences libres a été bâti sur celui du copyright américain. Celui-ci possède en effet un statut de *license* par lequel le titulaire de droits renonce publiquement à certains de ces droits. Ces *licences* sont des engagements unilatéraux de détenteurs de droits soumis à un régime fédéral dans le cadre des lois sur le copyright alors que le *contract* (similaire à notre notion française) est soumis au régime général des contrats. Les conséquences pratiques sont de plusieurs ordres : concernant le formalisme requis (une licence ne pouvant être contestée par le bénéficiaire qui se retrouverait sans droit, elles sont généralement beaucoup plus courtes que les contrats), la juridiction compétente ou la loi applicable (régime fédéral ou loi d'un État) ou encore la réparation attachée à leur violation (interdiction d'exploiter ou exécution forcée en cas de licences, simple dédommagement en cas de contrat). C'est pour toutes ces raisons que certains revendiquent la qualification de

loi du 3 juillet 1985 (complétée plus tard par la directive 91/250/CEE du 14 mai 1991) qui entérina cette position.

1. Unix était ainsi disponible pour les universités à 150\$ et à 20 000\$ pour les autres. Voir S. Holmgren, « Network UNIX », [RFC] : <http://tools.ietf.org/html/rfc681>, consulté le 8 octobre 2012.

*licenses*¹, même si un tel statut n'existe pas dans notre régime de droit latin (ni même, plus largement, en Europe)².

Près de trente années plus tard, les licences libres font l'objet d'une véritable attention, de la part tant du monde professionnel que des utilisateurs et du secteur public. Elles remettent en cause un certain nombre de modèles économiques, de manière d'autant plus forte que de réelles politiques publiques commencent à se formaliser en faveur du libre (au point où la pratique du logiciel libre dans certains marchés publics devient le principe) – ajoutons qu'en réponse aux plaintes de certains éditeurs traditionnels, menacés par la généralisation du logiciel libre, plusieurs décisions sont venues reconnaître la compatibilité (et les bienfaits) du logiciel libre au regard du droit de la concurrence³. Grâce aux efforts d'une multitude d'acteurs, cette composante contractuelle d'un système alternatif a réussi à intégrer au fur et à mesure les évolutions du système classique afin de s'appuyer sur ces derniers – voire les contrer lorsqu'ils faisaient obstacle aux objectifs initiaux.

Pour autant, l'objet de cet article n'est pas tant d'étudier les licences libres comme objets juridiques⁴, mais de comprendre, au travers d'une perspective historique, le fonctionnement de leur processus d'édiction (la forme) ainsi que l'évolution de leur contenu (le fond). Ainsi, complétant les analyses plus juridiques qui évaluent chaque licence au regard d'un droit positif (ensemble de règles juridiques en vigueur à un moment

1. À noter que les deux qualifications (*contract* et *license*) ont été retenues par les juges de l'affaire *Jacobsen v. Katzner*, United States Court of Appeals for the Federal Circuit, 18/08/2008 (<http://www.cafc.uscourts.gov/images/stories/opinions-orders/08-1001.pdf>). En Europe, les décisions allemandes (en 2004 contre Sitecom Deutschland, en 2006 contre D-Link Deutschland ou encore contre Skype en 2008) reposent toutes sur la reconnaissance de la GNU GPL comme un contrat valide permettant la sanction du non-respect du formalisme qu'elle impose.

2. La distinction entre contrat et *license* n'existe pas en droit romain. Un juriste français ne verrait ainsi dans la licence qu'une offre d'un auteur (ou titulaire de droits) se transformant en contrat dès l'acceptation par l'utilisateur. L'article 1101 du Code civil définit le contrat comme « une convention par laquelle une ou plusieurs personnes s'obligent, envers une ou plusieurs autres, à donner, à faire ou à ne pas faire quelque chose. »

3. Voir *Court of Appeals for the Seventh Circuit*, *Wallace v. International Business Machines Corp. et al.* (2006) ; *Cour de Munich*, *Welte vs. Skype Technologies SA* (2007) ; *Décision numéro 350431 du Conseil d'État français* (2011) ou encore la *Décision de la Cour constitutionnelle italienne* (2011).

4. Sur le sujet, voir Mélanie CLÉMENT-FONTAINE, « Les oeuvres libres », Thèse de doctorat, Université de Montpellier, 2006, p. 99 ; Benjamin JEAN, *Option libre. Du bon usage des licences libres*, Paris : Framasoft/Framabook, 2012.

donné), il s'agira ici d'apporter un éclairage utile à la compréhension des clauses et des mécaniques des licences libres.

Notre étude portera donc sur l'évolution des licences libres et *open source*¹, de la première formalisation de la notion de logiciel libre au début des années 1980 à nos jours, en portant une attention particulière aux licences structurantes du mouvement. Eu égard à l'objet, les mutations évoquées seront essentiellement juridiques, mais nous adopterons une vision pratique tout en gardant à l'esprit les objectifs initiaux (avoués ou non) afin de situer l'évolution des licences dans leur contexte. Une attention particulière sera apportée aux finalités des licences, celles-ci permettant d'en comprendre l'apparition de nouvelles, leur renforcement pour répondre aux besoins et enfin leur structuration dans une vision globale supplantant chaque vision individuelle.

1. La distribution des cartes : le processus d'apparition des licences libres

Un processus itératif. Il n'est pas étonnant, au regard de leur objet, que les licences libres soient conçues dans le cadre d'une démarche itérative et progressive. Elles évoluent au fur et à mesure que la société se transforme et suivent plus ou moins rapidement les évolutions du droit et de la technique. Une première conceptualisation des licences trouve son origine dans les vingt dernières années du XX^e siècle (1.1), entraînant l'arrivée d'une vague de nouvelles licences les dix années suivantes (1.2).

1.1 La conceptualisation des licences libres(1984-1998)

1.1.1 Les premiers logiciels « libérés » par leur licence (1984-1989)

Entre domaine public et licence libre. Conditionné au respect d'un certain formalisme, le bénéfice du copyright américain n'était pas automatique au début des années 1980. Ainsi, de nombreux projets logiciels furent codéveloppés, de manière collaborative et ouverte, avant que ne soit

1. On oppose aujourd'hui licences libres à logiciels propriétaires de la même façon que l'on oppose *open source* à *closed source*, sans que les licences libres et *open source* soient véritablement superposables. Voir *infra*.

expressément apposée une licence explicitant les droits accordés. Ce n'est que plus tard et en réaction à l'élargissement des communautés d'utilisateurs et contributeurs qu'il fut jugé utile de formaliser les interdictions associées à la réutilisation du code, dans un premier temps, et de formaliser plus précisément les droits accordés, dans un second temps. Cet éloignement du domaine public se fit avec douceur. L'exemple de Tex, système logiciel de composition de documents créé à partir de 1977, est symptomatique. Selon les dires de son auteur Donald Knuth, Tex était ainsi dans le domaine public avec comme seule contrainte (ajouté en 1982 lors de la sortie de TeX82) que « TEX » ou « METAFONT » ne soient pas réutilisés dans des versions dérivées du logiciel¹. Plus restrictif, Larry Wall (futur développeur du langage de programmation Perl) autorisait sur le logiciel *trn* tout usage dès lors que l'utilisateur ne prétendait pas être l'auteur du code ou qu'il ne cherchait pas à le vendre.

L'apparition des premières licences *copyleft*. Auteur du logiciel GNU Emacs, Richard Stallman lui associa en 1985² une licence spécifique intitulée la GNU Emacs General Public License : sensiblement plus longue que les licences précédentes, elle définissait avec précision le cadre permettant la réutilisation de la licence et obligeait à conserver la licence sur toutes les versions dérivées (ou contenant une partie) du logiciel. Cette protection découlait essentiellement de déboires que connut Stallman lorsqu'il souhaita réutiliser une version modifiée de son logiciel Emacs (Gosling Emacs) et qu'il se vit opposer le copyright de l'auteur de cette dernière – situation qui le contraignit finalement à tout réécrire pour ne pas être inquiété par la société qui avait acheté les droits. Conditionnant la licence (au sens de permission) à une certaine réciprocité, il se servit ainsi de ses propres droits d'auteur pour limiter toute réappropriation par un autre. Ressentant comme une menace la combinaison de code libre et non libre, cette licence formalisa la notion de *copyleft* par laquelle l'ajout ou la jonction d'un élément quelconque à un programme sous une telle licence doit se faire de sorte que la version résultant de cette opération soit régie par les mêmes termes (et donc également libre).

1. Donald E. KNUTH, « Computers and Typesettings », dans : *TUGboat* 7, 1986, p. 95-98.

2. Sam WILLIAMS, Richard STALLMAN et Christophe MASUTTI, *Richard Stallman et la révolution du logiciel libre. Une biographie autorisée*, Paris : Eyrolles, 2010, p. 126.

Les premiers logiciels universitaires. Parallèlement, et en réaction aux premiers courants de l'appropriation logicielle, les universités s'émancipèrent progressivement des éditeurs traditionnels tout en optimisant la diffusion (et la réutilisation) de leur logiciel par l'usage de licences libres. Elles firent ainsi du code la même chose que ce qu'elles faisaient de la recherche : le rendre disponible et réutilisable par un maximum de personnes. Cela fut réalisé dès les années 1986 pour le MIT (lors de la sortie de la version X10R3 du logiciel X Window System) ou un peu plus tard pour Berkeley qui devait se séparer de toutes les contributions d'AT&T afin de distribuer son système d'exploitation librement (BSD-Tahoe fut publié en 1988 et Net/1 en 1989). Répondant ainsi aux besoins qui lui avaient été exprimés (Stallman joua notamment un rôle important dans cette évangélisation), le monde académique et universitaire américain offrit ainsi à la société de nombreux logiciels qui composent aujourd'hui encore l'infrastructure d'Internet (on peut aussi citer le système de nom de domaine BIND, le protocole TCP/IP et Sendmail). Tous ces logiciels eurent pour point commun d'être diffusés sous des licences qui étaient courtes, organisant une cession des droits très large tout en excluant toute garantie ou responsabilité. La principale condition concernait la mention de l'origine du code, dans des termes variables. Par commodité, ces licences ont rapidement été nommées en référence aux projets éponymes : BSD license (plus tard dite *4-clause license* ou *original BSD License*) et MIT license.

La formalisation de la Free Software Definition (FSD). Fort du succès de GNU Emacs, Stallman fonda en octobre 1985 la Free Software Foundation (FSF), organisation américaine à but non lucratif, pour aider au financement du projet GNU et de la communauté du logiciel libre¹. C'est dans ce cadre que fut rédigée la Free Software Definition en 1986², cette dernière définissant comme libre tout logiciel pour lequel l'utilisateur bénéficiait de trois libertés : la liberté d'étudier le fonctionnement du programme, et de l'adapter à ses besoins (l'accès au code source et donc nécessaire) (liberté 1) ; la liberté de redistribuer des copies du logiciel (liberté 2) ; la liberté d'améliorer le programme et de publier ses améliorations.

1. Le projet GNU est officiellement annoncé par un message de Richard Stallman en date du 27 septembre 1983 ayant comme sujet « *new Unix implementation* ».

2. Richard M. STALLMAN, « What is the Free Software Foundation ? », dans : *GNU's Bulletin* 1 (1986), p. 8.

rations (là encore, l'accès au code source est nécessaire) (liberté 3). La liberté 0, celle d'exécuter le programme pour tous les usages, n'apparut que plus tardivement, aux alentours des années 1996, lorsque Stallman s'aperçut qu'elle pouvait ne pas être déduite des trois autres. C'est au regard de cette définition finaliste (recherchant la liberté de l'utilisateur final) que furent comparées les licences *copyleft* et les licences permissives : les licences permissives assurent ces libertés pour les premiers utilisateurs du logiciel seulement alors que les licences *copyleft* les rendent persistantes au profit de tous les utilisateurs (premiers et subséquents) du logiciel¹, faisant reposer cette charge sur les licenciés eux-mêmes (les astreignant par exemple au maintien de la même licence sur leurs créations dérivées)². Depuis lors, ces deux courants cohabitèrent sans heurt majeur.

De la GNU Emacs Public License à la GNU General Public License.

La GNU Emacs General Public License fut la première de la longue série des licences *copyleft* rédigée par la FSF : suivirent ensuite des licences similaires associées à la sortie du GNU Debugger (GDB) en 1986, du GNU Compiler Collection (GCC) en 1987, du GNU Bison en 1988, etc. Le texte initial fut adapté à chaque logiciel et évolua au cours de ces quelques années en fonction des retours au sein de la communauté : la seconde version de la GNU Emacs GPL introduisit par exemple en 1988 la notion de « sphère privée » (les obligations de la licence ne doivent être respectées qu'à partir du moment où le logiciel est distribué, sortant de cette sphère privée). D'un statut simplement déclaratif, la licence était devenue l'instrument qui définissait les « règles de jeux » nécessaires pour encadrer la collaboration au sein d'une communauté d'auteurs, utilisateurs et contributeurs. Cette traduction juridique des libertés définies prit la forme de cessions de droits en intégrant certaines obligations pratiques nécessaires pour que l'utilisateur jouisse effectivement de ces libertés (accès au code source dans sa forme préférentielle, sa documentation ou, dans une certaine mesure, la qualité du code). Ce premier travail reçut un accueil très favorable de la communauté que réunissait le projet GNU et plus largement (attentif au travail mené par la FSF, certains projets commencèrent ainsi à modifier ces licences afin de les utiliser à l'intention de leur propre

1. Richard M. STALLMAN, « The GNU Operating System and the Free Software Movement », dans : *Open Sources : Voices from the Open Source Revolution*, sous la dir. de Chris DiBONA, Sam OCKMAN et Mark STONE, Sebastopol, CA : O'Reilly, 1999.

2. JEAN, *op. cit.*, p. 201.

logiciel¹ et au-delà de la FSF – telle la Nethack General Public License dès 1989).

Genèse de la GNU GPL. Évoquée dès 1988, ce n'est qu'au début de l'année suivante (le 25 février 1989) qu'apparut la licence générique du projet GNU sous le nom de GNU General Public License (GNU GPL). Il s'agissait par ce texte de répondre aux besoins du projet GNU (notamment en termes d'harmonisation) et plus largement (puisque c'est aussi l'époque où d'autres communautés indépendantes commençaient à se former) avec pour volonté de favoriser la reprise de code entre les multiples projets libres qui se créaient. Cette décision fut stratégique vis-à-vis de l'utilisation et de la diffusion de la licence, la rendant rapidement centrale dans le paysage du Libre. Elle devenait ainsi plus simple à faire évoluer (puisque le rédacteur n'avait alors plus qu'un seul texte à modifier), plus simple aussi à comprendre (car plus utilisée – même si une FAQ et le *GNU manifesto*, datant de 1985, furent ajoutés pour mettre un terme aux divergences naissantes). Par la même occasion, un travail complémentaire fut aussi fourni afin de perfectionner la licence : en rallongeant le préambule de la licence ; en précisant les notions techniques (par exemple le code source qui fut défini comme la forme de création préférée pour faire des modifications sur celle-ci) ; en précisant les contraintes attachées au *copyleft* (notamment en formalisant l'interdiction d'ajouter des restrictions lors de la rediffusion). De multiples facteurs peuvent certainement expliquer son adoption rapide et large, mais la confiance particulière dont bénéficiait son rédacteur est certainement le plus important. Il n'est ainsi pas exagéré de considérer que ce que fit la FSF n'était à portée d'aucune autre organisation. Stallman, et avec lui la FSF, furent ainsi précurseurs dans la réunion de ce qui s'avéra plus tard être le terreau idéal pour le développement de logiciels libres : une communauté importante, des licences libres « génériques » et les bases d'une première gouvernance.

Une évolution parallèle à celle du copyright. L'histoire est une succession de coïncidences et d'intuitions. L'expansion d'Internet et du Web fut ainsi un accélérateur majeur dans la constitution des communautés de logiciels libres, de la même façon que le contexte politique est certainement

1. Stallman, et la FSF après lui, ont toujours été très permissifs à l'égard de leur licence en autorisant expressément toute réutilisation des clauses dans d'autres contrats dès lors que le nom et le préambule restaient inchangés.

corrélé à la rédaction de ces licences durant les années 1990. En effet, ces dates correspondent à l'époque où le régime du copyright américain subit une véritable rupture caractérisée par la ratification par les États-Unis de la convention de Berne le 16 novembre 1988¹, entraînant l'année suivante l'abandon du système du dépôt obligatoire au profit du régime de protection automatique (pratiqué depuis longtemps en Europe). Alors que la publication sans mention légale entraînait auparavant l'abandon de tout bénéfice du régime du copyright (l'enregistrement étant la condition pour défendre ses droits), l'ouverture et le partage de code devinrent dès cette date conditionnés à un comportement actif des auteurs se matérialisant par l'usage d'une licence (l'absence de licence étant un obstacle au partage et à la diffusion du code). Chacun put alors faire preuve d'originalité dans la rédaction de sa propre licence là où un comportement matériel (l'absence de mentions légales) suffisait auparavant. Il serait à cet égard intéressant de se demander si le logiciel libre aurait connu le même développement en l'absence de ce renversement de paradigme.

1.1.2 L'apparition des premières variations (1991-1998)

La réutilisation des licences. En raison de la grande liberté qu'elles accordaient, les premières licences BSD ou MIT se sont rapidement retrouvées dans une multitude de formes, fonctions de leurs utilisations : on repère ainsi des premières variations, qu'il s'agisse du projet Expat pour la licence MIT (par souci de précision, on distingue aujourd'hui chaque version de la licence initialement utilisée par le MIT : l'Expat License, la X10 et X11 License, etc.) ou des projets sous licence BSD (de la même façon, on distingue aujourd'hui les licences *BSD modified* ou *BSD simplified*). Par exemple, et pour faire un lien avec l'évolution d'Internet, Internet Systems Consortium (qui développe et maintient un certain nombre de projets essentiels au réseau : BIND, OpenReg, DHCP, etc.) distribua ses logiciels sous une licence ISC dérivée de la licence *BSD modified*, en s'appuyant sur les mécanismes introduits par la convention de Berne. Il s'agit d'une version simplifiée de cette dernière, et qui devint en 2003 la

1. Et son entrée en application aux États-Unis le 1^{er} mars 1989 par la « Berne Convention Implementation Act ». Voir *U.S. Copyright Office – Information Circular*, Copyright Notice 23.0811. Notons le vote en 1990 du Computer Software Rental Amendments Act, renforçant les prérogatives des ayants droit.

licence du projet FreeBSD. Cette tendance se confirma dans les années qui suivirent par l'adoption de telles licences par un certain nombre de projets sous le même nom où en s'appropriant parfois la licence (citons pour l'exemple la CWI License Agreement utilisée sur Python dès 1991 et directement inspirée de la licence type MIT, ou la Cryptix General License utilisée dès 1995 sur le projet Cryptix et directement inspirée de la BSD).

La déclinaison des licences GNU. S'appuyant sur les nombreux retours de la communauté croissante des hackers ayant adopté cette licence, une seconde version de la GNU GPL fut publiée en 1991 (introduisant pour la première fois une clause relative aux brevets) aux côtés d'une licence moins contraignante destinée à certains programmes qui gagnaient à être diffusés et utilisés en dépit d'un environnement non entièrement ouvert : la GNU Library General Public License (GNU LGPL). La GNU LGPL (immédiatement en version 2 aux fins d'harmonisation) autorise expressément la liaison de logiciels non libres avec la bibliothèque distribuée sous cette licence. À noter cependant que ces concessions n'étaient pas une renonciation aux libertés telles qu'elles avaient été formalisées au travers de la GNU GPL, mais un choix (initialement conçu pour la bibliothèque C) destiné à favoriser l'immixtion de logiciels libres stratégiques dans l'écosystème des logiciels propriétaires (dans les situations où, en présence d'alternatives, la licence aurait limité l'usage de ces logiciels) et, surtout, permettant aux éditeurs de développer des applications pour le système d'exploitation GNU sans les contraindre au choix de la licence GNU GPL (et donc rendre plus attirant ce dernier pour les développeurs)¹. Plus tard, en 1999, afin de freiner l'adoption systématique de la GNU LGPL pour les bibliothèques logicielles, la FSF modifia son nom au profit de GNU Lesser General Public License (la version devint alors 2.1) : une modification conforme aux objectifs qui avaient mené à la rédaction de la GNU LGPL, celle-ci n'étant stratégiquement intéressante qu'à l'égard de projets pour lesquels de grandes contraintes auraient eu

1. Par ailleurs, cette nouvelle licence répondait à la ligne de conduite que s'était fixée Stallman dès sa première licence libre. La GNU Emacs GPL précisant en effet dans son article 5 : « *If you wish to incorporate parts of GNU Emacs into other free programs whose distribution conditions are different, write to the Free Software Foundation. We have not yet worked out a simple rule that can be stated here, but we will often permit this. We will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software.* »

des effets trop néfastes sur leur diffusion (pour les autres, la GNU GPL devait être privilégiée). Les licences GNU GPL et GNU LGPL composèrent ainsi la première famille de licences : cohérente, complémentaire et solide juridiquement, celle-ci assurait la constitution d'un « pot commun » dans lequel toute personne était conviée à piocher dès lors qu'elle était prête à y reverser ses propres contributions. Encore peu juridiques¹, elles conservaient un langage très proche de celui des développeurs et étaient empreintes d'une intention forte qui se traduisait souvent par des termes complexes à définir (et donc à qualifier juridiquement).

Un système relativement complet. Le cumul des premières licences *copyleft* et des licences permissives donna un premier visage cohérent au système des licences libres. Ainsi, soit le contributeur souhaitait contribuer à un pot commun en usant de l'une des licences rédigées par la FSF soit il optait pour un partage plus large encore et se tournait vers l'une des licences rédigées par les universités. Dans les deux cas, les licences existantes pouvaient être utilisées en l'état – avec, bien heureusement pour les licences GNU, une aide communautaire concernant leur compréhension et usage. Les contraintes pour les développeurs étaient donc de trois ordres : tout reverser s'ils utilisaient un programme sous GNU GPL ; ne reverser que les modifications apportées à la bibliothèque sous GNU LGPL si leurs programmes dépendaient de celle-ci ; ou se limiter aux seules attributions obligatoires si le programme était sous une licence permissive (Apache, BSD ou MIT).

L'adoption de la GNU GPL par d'autres projets : Ghostscript, Perl et Linux. Qu'il s'agisse de L. Peter Deutsch, de Larry Wall ou Linus Torvald, ces trois développeurs de renom étaient convaincus de l'intérêt du Libre pour leurs projets (respectivement Ghostscript, Perl et Linux) : en termes de méthode de développement, de diffusion et extension. Approché et accompagné à ces débuts par Richard Stallman, L. Peter Deutsch avait logiquement choisi de diffuser Ghostscript sous GNU GPL dès son lancement dans les années 1988-1989. Une fois leurs projets matures, Larry Wall et Linus Torvald troquèrent plus tard leur propre licence par la GNU GPL : en 1989 pour le langage Perl (datant de 1987, et auparavant sous une licence interdisant de revendre le logiciel) et en 1992 pour

1. L'avocat et professeur de droit Eben Moglen ne rejoignit néanmoins cette cause qu'à partir des années 1993.

le noyau Linux (alors en version 0.99 – il était auparavant distribué sous une licence obligeant la fourniture des sources et interdisant l'usage commercial). Ces trois adoptions constituèrent un signal fort : représentatif de l'intérêt que présentait la GNU GPL à l'égard des projets collaboratifs, renforçant celui-ci du fait de leur légitimité propre. Cependant, elles illustrèrent tout aussi bien les premières frictions face aux positions très fortes que traduisait cette licence. Ainsi, Larry Wall et Linus Torvald ajoutèrent chacun une interprétation à la GNU GPL pour relativiser certains de ses effets en matière de combinaison de logiciels (cf. infra, section 2.1.1). Plus encore, Larry Wall rédigea en mars 1991 une nouvelle licence qu'il intitula Artistic License et L. Peter Deutsch créa en 1994 l'Aladdin Free Public License : l'Artistic License fut publiée à l'occasion de la sortie de la version 4 du langage Perl et s'ajouta à la licence GNU GPL (donc sous double licence) afin de « rassurer les industries du logiciel »¹ ; l'Aladdin Free Public License fut utilisée par la société Aladdin Entreprise pour interdire certains usages commerciaux, en complément de la version GNU Ghostscript (en retard de deux versions sur l'Aladdin Ghostscript) toujours diffusée sous licence GNU GPL.

De nouvelles licences permissives. En raison de leur simplicité de prise en main, de nombreuses nouvelles licences permissives apparurent durant les années 1990. La plupart étaient des variations plus ou moins fidèles des licences publiées par le MIT et Berkeley, mais d'autres se démarquèrent. La licence Zlib est apparue en 1995 en ajoutant quelques exigences quant à l'identification des modifications apportées. L'Apache Group, à l'origine du serveur web éponyme, basa ses développements sur le serveur httpd de la NCSA (sous une licence très libérale de type « *This code is in the public domain, and give us credit* ») et adopta en 1995 une première version de la licence Apache s'inspirant de la licence BSD originale (changeant le nom de l'organisation et ajoutant l'interdiction d'utiliser certains termes dans les produits dérivés).

1. Il tentait ainsi de répondre à un certain nombre de problèmes identifiés : en réduisant le *copyleft* (dès lors que le programme modifié n'entraînait pas en conflit avec le programme initial), en donnant la possibilité d'être partie d'un logiciel plus large (sans impact pour celui-ci quant à la licence) ; en supprimant tout impact du fait de codes laissés dans les logiciels générés.

Les complications de la clause de paternité. La première version de la licence BSD (dite *4-clause BSD*) contenait une clause qui devint rapidement polémique :

Tous les documents publicitaires mentionnant les caractéristiques ou l'utilisation de ce logiciel doivent afficher la mention suivante : « Ce produit contient un logiciel développé par l'Université de Californie, Berkeley et ses collaborateurs. »¹

La contrainte formalisée par cette troisième clause imposait donc un affichage proéminent qui assurait d'une part le respect la paternité de l'université de Berkeley et participait d'autre part à une valorisation importante de ses travaux grâce à cette « communication forcée ». Néanmoins, cette clause qui parut initialement anodine – voire légitime – devint rapidement une plaie lors de la réappropriation de cette licence pour d'autres logiciels : chacun requérant une mention spécifique lors de l'usage de sa propre contribution, il s'en est suivi des situations où le formalisme devint beaucoup trop lourd pour être raisonnable. S'appuyant sur des exemples de logiciels nécessitant la mention de plus d'une quarantaine d'auteurs, Stallman fit un lobbying important pour que cette clause (qu'il qualifia d'« odieuse clause ») soit supprimée. William Hoskins, directeur de l'Office of Technology Licensing for UC Berkeley fut sensible à ces arguments et publia une nouvelle version de la BSD (*new BSD* ou *modified BSD*) amputée de cette clause le 22 juin 1999. D'autres licences suivirent le mouvement : la Zope Public License (publiée en 1999) supprima une clause similaire en 2001 lors de son passage en version 2 ; la licence Apache fit de même en 2000 (passant ainsi en version 1.1). Malgré cela, de nombreux composants restèrent soumis à ces termes initiaux, obligeant le maintien de cette obligation au sein des programmes les réutilisant, telle la Sendmail License (2004) qui mêla un *copyleft* et des engagements hérités de la BSD originale (citation de l'université de Californie) au travers d'un article 5 qui y fait expressément référence ; ou encore la licence OpenSSL (1998) qui maintient les obligations héritées de la licence Apache initiale à laquelle s'ajoutait la licence SSLeay (*4-clause BSD*) sous la forme de ce qui fut appelé *dual-license* (qui, dans ce contexte, signifiait que l'utilisateur devait respecter les deux licences à la fois).

1. Traduction libre.

1.2 La ruée vers l'*open source* et la multiplication des licences (1998-2007)

L'apparition de nouvelles licences. Le XX^e siècle déboucha sur la reconnaissance du concept de logiciel libre en termes de réalité économique et technique. Il marqua l'entrée dans une courbe de croissance continue. L'arrivée de nouvelles licences fut peut être le corollaire imprévu, induit par la reconnaissance de nouveaux besoins, l'arrivée de nouveaux acteurs cherchant à se rattacher au mouvement en construction ou encore l'évolution des projets initiés dans les années 1980. Le premier lot de licences témoignait d'une faible présence européenne ou asiatique sur le sujet (même si de nombreux projets avaient déjà une dimension mondiale). Le début du XXI^e siècle, quant à lui, introduisit la diversité avec l'apparition de nouvelles licences, créées par des personnes publiques ou privées, plus seulement américaines, et dépassant le seul secteur du logiciel. Ce fut ainsi à la fois une réussite en termes d'expansion du phénomène (alors qu'à la même époque était voté le Digital Millennium Copyright Act¹ aux États-Unis), mais aussi un échec en termes d'harmonisation puisqu'on ne compta plus le nombre de nouvelles licences qui vinrent compléter les licences existantes (elles-mêmes souvent modifiées).

Des licences parfois inadaptées. Rapidement, les licences furent confrontées à leurs propres limites : l'effet relatif et la prévisibilité des contrats. L'effet relatif est la limite la plus importante du système : seules les personnes qui sont parties aux contrats sont engagées par ce dernier dans les limites de ce qu'elles ont convenu, de telle sorte que, sauf stipulations contraires, le rédacteur du contrat est tiers à cette relation et ne peut imposer d'aucune manière certains usages ou même certaines interprétations. Même s'ils peuvent parfaitement être utilisés pour encadrer des relations sur du très long terme, les contrats ne restent pertinents que s'ils réussissent à encadrer la relation telle qu'elle avait été prévue initialement, les procédures de révisions et d'adaptation des contrats étant alors généralement fastidieuses. Ainsi, la généricité des licences initiales, si elle

1. Votée en 1998, cette loi introduisit notamment la protection juridique au profit des mesures techniques de protections. Son équivalence européenne est l'EUCD (directive 2001/29/CE du Parlement européen et du Conseil du 22 mai 2001) et transposée en France par la loi DADVSI (Loi n°2006-961 du 1^{er} août 2006 relative au droit d'auteur et aux droits voisins dans la société de l'information).

a sans doute facilité leur réutilisation, n'a pas suffi pour empêcher l'apparition de nouvelles licences destinées à répondre à des situations particulières (communautés, domaine, contributeurs types, langage, voire objet de la licence lorsqu'on dépasse la sphère du logiciel). Fort heureusement, un rôle structurant essentiel à l'écosystème fut assuré par les organisations qui régulèrent et accompagnèrent l'apparition et l'usage des différentes licences libres. Ces organisations furent d'autant plus importantes dans la phase suivante que l'intérêt (et l'utilité) de nouvelles licences est très vite apparu comme relevant de deux conditions : 1) provenir d'une organisation bénéficiant d'une certaine légitimité (ou créées à cette fin et héritant de la légitimité de ses membres) et 2) répondre aux besoins d'une communauté (structurée ou non).

Deux mouvements parallèles caractérisent cette décennie : la structuration du secteur du logiciel avec l'industrialisation des licences libres (1.2.1) et l'extension du Libre à de nouveaux domaines (1.2.2).

1.2.1 L'arrivée de nouvelles licences libres industrielles : la phase de diffusion et d'émancipation

Le succès des licences *copyleft*. Concernant le type de licences, on remarque que la plupart des licences rédigées pour des projets d'origine privée (sociétés ou associations) étaient *copyleft* alors qu'à l'inverse les projets scientifiques ou communautaires privilégiaient les licences permissives, tel le W3C qui rédigea sa propre licence fin 2002 sous le nom de Software Notice and License. Les premiers cherchaient ainsi à maintenir un contrôle en empêchant la réappropriation de leur création alors que les projets tournés uniquement vers une optimisation de la diffusion s'orientaient vers des licences permissives. Tout à fait logiquement, compte tenu de la longueur et de la complexité des textes, les améliorations concernaient principalement les licences *copyleft* et étaient généralement destinées à renforcer la pérennité recherchée par l'usage de la licence : par une meilleure formulation de la licence relative aux brevets, par la recherche d'un élément déclencheur adapté, etc. À l'inverse, les seules améliorations constatées en matière de licences permissives consistèrent au raccourcissement des textes : telle la Fair License rédigée en 2004, qui est la plus courte des licences certifiées par l'OSI (composée de seulement avec les deux phrases), ou la Wide Open License (WOL).

L'attrait de l'*open source* pour la diffusion des standards. Le développement du marché de l'édition logicielle a entraîné très rapidement des positions de monopoles et de domination par certains éditeurs. En réaction, l'industrie du logiciel a rapidement compris l'intérêt que présentait l'*open source* dans la (re)conquête d'un secteur ou d'un marché. Les sociétés s'accordèrent aussi sur l'intérêt crucial de la compatibilité et de l'interopérabilité. La Sun Industry Standards Source License (SISSL) fut utilisée par OpenOffice.org jusqu'en 2005. Elle permettait la réutilisation du code libre et sans contrainte dès lors que les modifications ne déviaient pas du standard (les APIs OpenOffice.org et le format de fichiers OpenOffice.org utilisé par OOo) et devenait *copyleft* en cas de déviation trop importante. L'adoption des libertés du logiciel libre ne fut pas instantanée et il est intéressant de constater l'évolution du positionnement des sociétés privées qui cherchèrent initialement à proposer des équilibres conçus à leur avantage (la Sun Community Source License publiée en 1998 différenciait ainsi les usages pour la recherche, l'utilisation/déploiement et l'utilisation commerciale) et optèrent progressivement pour des licences de plus en plus équitables, répondant aux définitions du logiciel libre et de l'*open source*. Les sociétés eurent donc tôt fait de comprendre et de se plier aux attentes des communautés qu'elles sollicitaient.

L'impact de l'industrialisation sur les licences. Bénéficiant ainsi de retours précieux (en termes d'utilisation et de développement) de ses utilisateurs, l'industrie du logiciel investit de plus en plus au profit de cette innovation partagée. Cela eut un impact fort sur le monde des licences, puisque l'immixtion de l'industrie passa par la mise à disposition des services juridiques des grandes sociétés de l'époque (IBM, Sun, Apple, etc.). La pratique juridique accessoire (et parfois amateur) devint ainsi une pratique stratégique (la gestion de la propriété intellectuelle étant une priorité pour ces sociétés éditrices ou intégratrices) et différentes modifications furent apportées : modifications accessoires (l'Intel Open Source License fut publiée en 2000 pour ajouter un article à la BSD quant aux lois relatives à l'export) ou plus importantes. À cet égard, la contribution prépondérante est sans aucun doute la rédaction de la Mozilla Public License (MPL) par la firme Netscape.

La Mozilla Public License (MPL). La licence Mozilla Public License (MPL) fut créée pour la diffusion du code source du Navigateur Netscape

Communicator de la société Netscape¹. La société souhaitait favoriser la création de solutions (même propriétaires) intégrant son logiciel tout en s'assurant que les modifications apportées à celui-ci resteraient accessibles et disponibles. En l'absence de licence adaptée, la rédaction d'une nouvelle licence fut décidée. Au confluent de l'industrie et des communautés, elle pouvait s'appuyer sur les licences existantes ainsi que sur les pratiques professionnelles des juristes de la société (et notamment Mitchell Baker, qui fut plus tard présidente de la Mozilla Foundation et de la Mozilla Corporation). La licence initialement soumise aux commentaires était la Netscape Public License (NPL), néanmoins, la prise en compte des multiples commentaires aboutit à la rédaction d'une nouvelle licence intitulée Mozilla Public License. En effet, la NPL garantissait à la société Netscape la faculté de relicencier sous toute autre licence (même commerciale) les contributions au code : une prérogative généralement conservée par le titulaire de droits et qui créait un déséquilibre au seul profit de Netscape, ce qui était mal accepté par les personnes consultées. Pour respecter l'ensemble des intérêts en présence, tout le code publié par Netscape le 31 mars 1998 fut publié sous NPL et tout développement subséquent sous MPL (avec une transition vers la MPL projetée en 2 ans).

La fondation de l'OSI. L'Open Source Initiative (OSI) fut imaginée le 3 février 1998 à Palo Alto, peu de temps après la libération du code de Netscape. Elle résultait du sentiment que Netscape ouvrait la porte à une dynamique plus grande, que se généralisait un mouvement d'ouverture aux communautés et qu'il devenait opportun de promouvoir cela en s'émancipant de la notion trop politique (et trop peu industrielle à leur goût) de logiciel libre. Le logiciel libre suivait encore les pas d'Internet : de la même manière qu'il avait imprégné l'infrastructure d'Internet lors de sa construction, il se banalisait en même temps que celui-ci. Fondée par Eric Steven Raymond et Bruce Perens, l'OSI est une *non profit public benefit corporation* (sorte d'association à but non lucratif) créée avec le soutien et l'étroite collaboration d'organisations de l'époque (telles que l'Internet Engineering Task Force ou l'Internet Systems Consortium qui voyaient dans l'*open source* un développement structurant pour Internet). Elle compta dès ses origines Todd Anderson, Chris Peterson (qui proposa l'appellation *open source*), John Maddog Hall, Larry Augustin, Sam

1. À ce sujet, voir l'article de Robert Viseur, dans cet ouvrage.

Ockman. Parmi ses premières contributions, l'OSI rédigea l'Open Source Definition (OSD) en s'appuyant sur les travaux récents réalisés en 1997 par Bruce Perens sur les Debian Free Software Guidelines (DFSG) du *Contrat Social Debian* (Debian est à cet égard le premier projet à avoir formalisé un « contrat social » assurant la pérennité du projet et une politique en matière de gestion de licence au sein de la distribution afin que ne soit inclus que des outils conformes à la philosophie GNU). Contrairement à la Free Software Definition, l'OSD s'intéresse essentiellement au contenu de la licence et les dix critères qu'elle énonce¹ déterminent les droits et conditions que doivent remplir les licences pour permettre un développement *open source*². Sur la base de l'OSD, l'OSI a accompagné de nombreuses entreprises et individus dans la rédaction de leurs licences, corrigeant leur rédaction et les accompagnant dans leur effort d'ouverture en certifiant leur licence d'*open source*³. Les conditions sont ainsi : la libre redistribution du logiciel (la licence ne peut, par exemple, exiger le paiement d'une redevance supplémentaire) ; la fourniture (ou l'accès) du code source ; la possibilité de créer des œuvres dérivées ; la préservation de l'intégrité du code (pour éviter l'appropriation par un tiers de contribution d'un autre – les modifications peuvent n'être éventuellement distribuées que sous forme de *patch*, séparément) ; l'absence de discrimination entre les personnes ou les groupes (toute personne détentricice d'une copie du logiciel bénéficie des termes de la licence tant qu'il s'y conforme

1. La dixième clause a été ajoutée en 2004 afin de s'assurer que les licences n'étaient pas spécifiques à un type de logiciel et de sortir de l'usage des *click-wrap*, le contrat n'étant conclu qu'à raison d'un clic de confirmation de l'utilisateur.

2. Il faut néanmoins reconnaître que les licences conformes à l'une des définitions le sont quasi systématiquement à l'autre, mais on peut trouver des exemples contraires, labellisées *open source*, mais non considérées comme libres, telles la Reciprocal Public License, l'Artistic licence, l'Apple Public Source License v1.x ou la Common Public Attribution License ; ou inversement telle la première licence de l'Affero GPL ou toutes les licences non encore labellisées telles que les CeCILL.

3. À noter que l'une des missions de l'organisation était de promouvoir la marque « Open Source ». Celle-ci ne fut néanmoins jamais détenue par l'OSI (une tentative d'enregistrement échoua en 1999 car le signe fut, déjà, considéré comme trop descriptif – pour éviter toute appropriation du langage, un signe ne peut en effet être accepté comme marque que s'il est suffisamment distinctif des produits ou services désignés : c'est-à-dire porteur d'un nouveau sens, ce qui n'était ici pas le cas puisque la notion d'*open source* était déjà largement reprise – contrairement aux autres marques : OSI, Open Source Initiative et les logos. Ce sont ces dernières qui permettent à l'OSI de contrôler l'usage « OSI Certified » par les éditeurs de logiciels *open source* et aussi sur cette base qu'a été conclu en 2012 un accord avec l'Open Source Hardware Association (OSHW) pour l'exploitation de son propre logo.

lui-même) ; l'absence de discrimination entre les domaines d'application (la licence se limite à la propriété intellectuelle : elle ne peut en aucun cas réguler d'autres champs de la société) ; l'absence de dépendance vis-à-vis d'autres contrats (par exemple un accord de confidentialité) ; l'absence de dépendance vis-à-vis d'un produit (la licence est attachée au code et non à un logiciel particulier) ; l'absence de contraintes vis-à-vis d'autres logiciels (ce qui ne s'oppose pas aux licences *copyleft* ayant un large *copyleft*, telle la GNU GPL qui ne s'étend qu'au programme envisagé comme un tout) ; la neutralité technologiquement. À l'instar de la FSD, l'OSD fut ensuite dérivée afin de transposer le courant du Libre et de l'*open source* dans d'autres domaines.

Les descendances de la MPL. Cette licence arriva rapidement aux oreilles des sociétés du secteur, attentives au mouvement naissant. La licence était spécifique au projet Mozilla, mais suggérait la création de licences dérivées : ce que firent très rapidement des sociétés sensibles à l'*open source* comme l'étaient Apple, Sun ou IBM. Bien d'autres licences s'en inspirèrent : telles la Nokia Open Source License (NOKOS License), l'Australian National University Open Source (ANUOS) License, CUA Office Public License, Interbase Public License, Ricoh Source Code Public License, Enhydra Public License, etc. Certaines y ajoutèrent des conditions supplémentaires telles la Yahoo ! Public License qui se réservait le droit de rompre le contrat si elle constatait une violation par l'utilisateur ou la Zimbra Public License qui étendit les obligations en termes de maintien des mentions légales aux mentions figurant dans l'interface utilisateur. L'université de Victoria publia en 2003 l'Adaptive Public License, une licence flexible basée sur la MPL qui donne au concédant le choix : de la juridiction, d'une licence de brevets, d'une extension aux usages internes, d'une formalisation des distributions de modification, et certaines attributions complémentaires. Très actif en matière d'*open source*, le NASA's Goddard Space Flight Center rédigea en 2004 sa propre licence en 2004 (NASA Open Source Agreement 1.3 – les versions précédentes n'ont jamais été publiques) basée sur la MPL et ses variantes.

L'apport d'IBM, Apple et Sun. Ces trois grands groupes surent très rapidement tirer profit de ce courant, mais ils y contribuèrent tout autant (reprenant et améliorant la MPL). Ainsi, IBM publia en 1999 l'IBM Public License en introduisant une notion de responsabilité à l'encontre de

l'entité (*publisher or distributor of the licensed software code*) qui prenait la charge de la distribution, en renforçant les clauses en matière de brevets (la licence étant annulée en cas de litiges de brevets) et en assouplissant le *copyleft* de la licence (afin de permettre l'ajout de liens). Elle inspira elle-même d'autres licences (notamment la Lucent Public License en 2000) et devint Common Public License en 2001 afin de faciliter sa réutilisation. La licence laissa la place à l'Eclipse Public License (EPL) en 2004, version relativement similaire modifiée en faveur de la fondation Eclipse. L'Apple Public Source License (APSL) aussi publiée en 1999 descendit d'un cran l'élément déclencheur : toute utilisation, même interne, emportait des obligations en termes de formalisme ; ajout d'une obligation de prévenir la société à chaque déploiement d'une version modifiée ; et une clause de résiliation arbitraire permettant à Apple de rompre la licence en cas d'accusation de violation de brevets ou de copyright. Après deux mises à jour en 1999 et 2001, toujours insuffisante aux yeux de la FSF, une version 2.0 a été publiée en 2003 avec le soutien de cette dernière. Sun publia en 1999 la Sun Public License (SPL) qui se contentait de remplacer les références à Mozilla par celles de Sun et, fin 2004, une licence plus aboutie sous le nom de Common Development and Distribution License (CDDL) qui complétait la MPL concernant la menace des brevets et, surtout, offrait la possibilité d'être réutilisée par tout autre projet (OpenSolaris, Glassfish).

La Qt Public License. L'évolution de modèles de licence autour de la bibliothèque Qt est intéressante en termes de dynamique. Co-fondateurs de TrollTech, Haavard Nord et Eirik Chambe-Eng développèrent en 1991 la bibliothèque Qt qu'ils commercialisèrent différemment selon qu'il s'agisse de la version Windows (Qt/Windows) ou Unix (Qt/X11). La version Unix était sous une licence Freeware (FreeQt licence) permettant le développement d'applications autour de celle-ci sur la plate-forme Unix alors que la version Windows ne pouvait être utilisée qu'à condition d'acheter une licence commerciale de la bibliothèque. Suite à la publication, en juillet 1998, de l'environnement de bureau K Desktop Environment (KDE) basé sur la bibliothèque Qt, la société Trolltech subit une forte pression de la communauté pour diffuser Qt sous une licence libre ¹.

1. Voir notamment « Position de Debian concernant les droits d'auteur et les problèmes de licence concernant KDE », 8 octobre 1998, URL : <http://www.debian.org/News/1998/19981008.fr.html>.

Elle prit celle-ci en compte par une double contribution : la rédaction de la licence Q Public License (QPL) pour la version Unix¹ et la création de la KDE Free Qt Foundation avec pour mission d'assurer la disponibilité et la pérennité de Qt (une convention lie l'éditeur à la communauté de telle sorte que s'il renonce à la publication de Qt sous sa licence *open source*, alors la fondation pourra diffuser Qt sous licence de type BSD ou toute autre licence *open source* de son choix). La licence Qt n'étant pas compatible avec la GNU GPL, la société Trolltech a modifié sa licence au profit de la GNU GPL v2 en 2000 (et GNU GPL v3 en 2008). En 2008, Trolltech est racheté par Nokia qui diffuse sous GNU LGPL 2.1 début 2009 et revend le projet à Digia le 9 août 2012.

L'Open Software License (OSL) est une licence issue des travaux de Lawrence E. Rosen, à l'époque conseiller juridique de l'OSI (avocat très impliqué dans l'*open source*, il fut directeur de la fondation Apache de 2011 à 2012). La première version de cette licence date de 2002. Modifiée la même année (version 1.1), Larry Rosen introduisit la notion d'*external deployment* (l'utilisation et les liaisons par réseau de logiciels) comme élément déclencheur afin de s'assurer que l'usage d'un logiciel par le réseau emporte les mêmes conséquences qu'une distribution de ce logiciel à l'utilisateur final. La version 1.1 fut certifiée *open source* par l'Open Source Initiative (OSI), mais fut déclarée incompatible avec la DFSG par la communauté Debian en raison de la clause de résiliation en matière de brevet. La version 2.0 date de 2003, et la dernière version, la troisième, date de 2005 (elle prend notamment en compte les remarques de la Commission européenne). Dès sa publication, la licence est expressément présentée comme un contrat (par opposition à la GNU GPL qui se présente comme une licence). Marginalement utilisée, elle est cependant très bien rédigée et a inspiré un certain nombre de licences (dont la *Jabber Open Source License*). Elle existe aussi dans une version permissive, sous l'appellation d'Academic Free License (aussi publiée en 2002). Ces licences furent un temps qualifiées de *best practice* par l'OSI (mi-2006) avant d'être catégorisées comme « redondante avec des licences plus populaires » quelques années plus tard.

1. La version Windows n'était toujours pas libre, le projet KDE sur Cygwin a travaillé en 2002 au portage de la bibliothèque Unix sur Windows et ce n'est qu'à partir de Qt/Windows 4, en juin 2005, que Trolltech a décidé d'étendre la GNU GPL à toutes les plates-formes (ce qui permit alors le développement d'applications multiplateformes).

L'évolution itérative des licences. Profitant de cette mouvance, le projet LaTeX s'est doté, début 1999, d'une nouvelle licence intitulée LaTeX Project Public License. Cette licence et ses versions 1.1 et 1.2 publiées la même année étaient tout à fait adaptées à la diffusion de la technologie LaTeX puisqu'elles permettaient la redistribution sans restriction de l'ensemble (mais d'une partie) des fichiers non modifiés, cependant elles encadraient beaucoup plus strictement la distribution de fichiers modifiés : celle-ci devant se faire, sauf mention contraire, dans l'en-tête des fichiers, à condition que ceux-ci soient renommés et que les auteurs initiaux soient clairement attribués. La licence fut modifiée en décembre 2003 avec le concours de Debian et de l'OSI, afin d'atténuer ce mécanisme en recommandant de tels changements de noms tout en permettant alternativement une identification « claire et non ambiguë » des versions modifiées (un mécanisme et une évolution pouvant ainsi être rapprochés de ceux des licences Artistic et Zope). L'Artistic License a aussi été modifiée à plusieurs reprises au profit de variations telles que la Simplified Artistic License, conçue en 2002 pour l'eNetwizard Content Application Server, ou encore la Clarified Artistic License, rédigée courant 2000 par Bradley Kunht (cette dernière sera en partie reprise lors de la publication en 2006 d'une version 2 de la licence).

Quelques licences particulières et non libres. L'année 2002, qui fut celle de la déclaration de guerre des USA contre le terrorisme, eut un impact sur les licences avec l'apparition de l'Hacktivismo Enhanced Source Software License Agreement (HESSLA – qui inspira le Torrifly Ethical Software License Agreement quelques années plus tard) ou encore la JSON License. La première est une licence relativement complète qui interdit les utilisations ou modifications qui violeraient les droits de l'homme ou entraîneraient l'espionnage des utilisateurs ; la seconde est une licence MIT à laquelle l'auteur (Douglas Crockford) ajouta « *The Software shall be used for Good, not Evil* ». Ces deux licences posèrent un certain nombre de difficultés au regard de leur subjectivité inhérente, de sorte qu'elles furent considérées comme « non libres » alors même que les logiciels ainsi développés présentaient un fort intérêt. Précisons qu'à la demande de la société, Douglas Crockford accepta d'accorder l'exception suivante : « *I give permission for IBM, its customers, partners, and minions, to use JSLint for evil.* »

Les licences « Badgeware ». Le terme *badgeware* apparut pour décrire ces licences qui imposaient l’affichage permanent de l’attribution des origines (telle la licence Zimbra) ou la Common Public Attribution License (publiée par la société Socialtext en 2007) qui permettait au développeur original d’imposer à chaque lancement du logiciel un affichage proéminent des attributions.

La seconde version de la licence Apache. En 2004, une refonte globale de la licence donna une version 2 beaucoup plus complète et rigoureuse répondant ainsi aux besoins de sécurité juridiques exprimés par les utilisateurs industriels. La fondation Apache considéra rapidement sa licence comme compatible avec la GNU GPL, mais cette compatibilité ne fut en réalité atteinte qu’à travers la version 3.0 de la GNU GPL.

Les licences CeCILL. En France et à l’initiative des principaux centres de recherche publique, une première licence libre *copyleft* française, conforme au droit français et européen (la Loi Toubon contraint les personnes publiques à contracter en Français en France), et compatible avec la licence GNU GPL, fut publiée le 21 juin 2004 sous le nom de CeCILL pour Ce(a) C(nrs) I(nria) L(ogiciel) L(ibre)¹. Ce travail avait été soutenu par Renaud Dutreil, alors ministre de la Fonction publique et de la réforme de l’État, qui était allé jusqu’à demander à l’Agence pour le développement de l’administration électronique (ADAÉ) « en concertation avec les trois organismes à l’origine de CeCILL, de porter ce travail au niveau européen et de le soutenir auprès des pays francophones ». Cette première version essuya néanmoins un certain nombre de critiques de la part de la FSF et de l’APRIL², de sorte qu’une seconde version vint en remplacement le 21 mai 2005 afin d’étendre la cession en matière de brevets, préciser certaines notions techniques ou formulations, et accorder une pleine valeur juridique à la version anglaise. Depuis, deux variations, nommées CeCILL-B et CeCILL-C, vinrent s’ajouter en 2006 à la « famille des licences CeCILL » dorénavant composée d’une licence permissive (CeCILL-B), d’une licence *copyleft* à l’étendue inspirée de la MPL (CeCILL-C), et d’une licence *copyleft* inspirée de la GNU GPL (CeCILL ou CeCILL-A).

1. Initiative comparable à la German Free Software License (Deutsche Freie Software Lizenz), elle aussi publiée en 2004, compatible avec la GNU GPL et disponible en allemand et anglais.

2. Association pour la promotion et la recherche en informatique libre.

Les licences *shared source* de Microsoft. Microsoft a formalisé dès 2001 un programme *shared source* qui ouvre dans une certaine mesure (en fonction des produits et des règles d'éligibilité prédéfinies) du code de la société à titre de référence et d'interopérabilité, pour l'étude de la sécurité de certains systèmes ou pour le développement. Plusieurs licences ayant été rédigées dans ce cadre, certaines avaient été soumises à l'OSI dès 2005 sans l'accord de Microsoft qui, après avoir été contacté, préféra reporter cette décision. C'est donc en 2007 que Microsoft proposa à l'OSI deux des licences de son programme Shared Source Initiative. Elles furent labellisées *open source* le 12 octobre 2007 et changèrent de nom à cette occasion : la Microsoft Public License (Ms-PL, auparavant Microsoft Permissive License) et Microsoft Reciprocal License (Ms-RL, auparavant Microsoft Community License). Les licences restantes (Microsoft Reference Source License, Ms-RSL, Microsoft Limited Public License, Ms-LPL, Microsoft Limited Reciprocal License, Ms-LRL) ne peuvent néanmoins pas prétendre à la qualification d'*open source*.

L'European Union Public Licence (EURL). La licence de l'Union Européenne (EURL) est apparue sous forme de brouillon en 2005 (v.0.1) et 2006 (v.0.2) avant d'être approuvée par la Commission Européenne le 9 janvier 2007. Disponible en 22 langues, chacune de ses traductions ayant la même valeur contractuelle, l'European Union Public License est sans conteste la licence la plus internationale. Il est intéressant de noter qu'une étude relativement poussée des licences existantes a précédé la rédaction de celle-ci, surtout en raison du faible nombre de licences européennes et des garanties souhaitées pour la commission (la version 3 de l'OSL avait pour objectif de corriger les défauts identifiés par la Commission afin de la rendre éligible). Une version 2.0 est discutée depuis fin 2012 afin de mettre à jour les termes initiaux et agrandir la liste des licences *copyleft*.

La structuration de l'*open source*. En matière de licence, les actions combinées du projet Debian, de l'OSI et de la FSF favorisèrent autant qu'elles encadrèrent cette prolifération. Néanmoins, de nombreuses autres organisations apparurent dans les années 2000 et eurent un rôle déterminant dans l'adoption et la consolidation du mouvement : l'association Software in the Public Interest (1997) les fondations Apache (1999), GNOME (2000), Mozilla (2003) ou Eclipse (2004) l'Open Source Development Labs (OSDL) qui devint la Fondation Linux en 2007 (après la

fusion avec le Free Standards Group, FSG). La plupart de ces organisations étaient dédiées à un projet en particulier, mais avaient généralement d'autres objets plus larges (apporter le soutien à d'autres projets Libres pour la fondation Apache, « préserver le choix et l'innovation sur l'Internet »¹ pour la fondation Mozilla, etc.).

1.2.2 L'extension à de nouveaux domaines

Un contexte particulier². De façon plus ou moins liée au mouvement du logiciel, les autres domaines de la création furent progressivement touchés. La FSF ne s'est pas clairement positionnée sur le sujet³, mais elle accompagna – par la personne de Stallman pour un premier temps, mais ensuite avec d'autres de ses acteurs – certaines de ces démarches. En parallèle et pour contextualiser ce phénomène, il faut se rappeler que cette époque est aussi celle des lois poussées par l'industrie culturelle : le Traité de l'OMPI sur le droit d'auteur de 1996, les lois américaines tels que le Sonny Bono Copyright Term Extension Act (étendant la durée des droits à 90 ans pour les créations salariales – aussi appelé « Mickey Mouse Protection Act » – Lawrence Lessig s'était illustré en contestant cette Loi devant la Cour suprême des États-Unis⁴) et le Digital Millennium Copyright Act (DMCA) de 1998 ; la Directive européenne sur l'harmonisation de certains aspects du droit d'auteur et des droits voisins dans la société de l'information (2001) transposée en France par la Loi DADVSI (Droit d'Auteur et Droits Voisins dans la Société de l'Information).

Les licences destinées à la documentation. La documentation associée au logiciel fut le premier type de création non logicielle pour lequel fut ressenti le besoin d'utiliser aussi une licence libre. Les motifs, simples, étaient que la documentation est l'un des facteurs clés de la collaboration et qu'il est donc nécessaire qu'elle soit aussi libre que le logiciel sur lequel elle portait. Jusqu'aux années 1998, les licences des logiciels étaient

1. Voir *The Mozilla Manifesto*, URL : <http://www.mozilla.org/about/manifesto.html>.

2. Pour cette section, et notamment pour une présentation plus précise de l'appréhension du Libre dans le domaine de l'art et du contenu, voir l'article de Primavera De Filippi et Isabelle Ramade, dans cet ouvrage.

3. Voir « Licences pour les œuvres d'opinion et de jugement », URL : <http://www.gnu.org/licenses/license-list.fr.html>.

4. Lawrence LESSIG, « How I Lost the Big One », dans : *Stories, Legalaffairs* (2004), URL : http://www.legalaffairs.org/issues/March-April-2004/story_lessig_marapr04.msp.

simplement étendues à leur documentation et il existait donc, à l'instar du logiciel, des partisans de la GNU GPL ou de la licence BSD (Expat). Néanmoins, l'importance de la documentation s'accrut au fur et à mesure que les communautés s'élargissaient et c'est donc dans le même temps que le logiciel libre gagna ses lettres de noblesse dans l'industrie du logiciel et que les licences furent rédigées en faveur de l'édition de documentation libre et ouverte – la Free Document Dissemination License (FDDL) utilisée dès 1997 par l'INRIA favorisait la diffusion et l'amélioration (par le biais de notes et commentaires) sans faire de celle-ci une œuvre collaborative. Parallèlement, et conscient que la GNU GPL ne répondait pas parfaitement aux besoins, Stallman publia le 12 septembre 1999 une première version 0.9 « brouillon » de la GNU Free Documentation License, une licence dédiée à la documentation logicielle et conforme à la FSD. Après quelques révisions, une version 1.1 fut publiée en mars 2000 (la version 1.2 date de 2002). La documentation s'envisageant comme un tout, la licence devait étendre le *copyleft* initial à tout ajout ou modification réalisé sur la documentation initiale. Enfin, l'un des autres intérêts était le vecteur de communication que représentait la documentation : il était en effet possible d'expliquer, voire d'approfondir, au sein de cette documentation la démarche liée à l'utilisation d'une licence libre. Non satisfait de cette licence qu'elle trouvait trop complexe, Apple publia sa propre licence *copyleft* dédiée à la documentation un an plus tard : la Common Documentation License. Directement inspirée de la MPL, la Public Documentation License (PDL) est publiée en 2002 par Sun pour la documentation de certains de ses logiciels (notamment OpenOffice.org).

Le mouvement de l'Open Content. Dès juillet 1998, David A. Wiley publia l'Open Content License. Il s'agissait d'une licence *copyleft* relativement courte et permissive qui permettait à l'auteur d'interdire les créations dérivées (de sorte qu'il existait 4 variantes : OPL, OPL-A, OPL-B et OPL-AB : l'option A interdisait tout usage commercial de l'oeuvre, l'option B les modifications). La licence fut remplacée onze mois plus tard par l'Open Publication License, notamment utilisée par O'Reilly pour l'ouvrage d'Eric S. Raymond, *The Cathedral and the Bazaar*. Le projet *Open-Content* fut officiellement fermé le 30 juin 2003 lorsque David A. Wiley rejoignit Creative Commons comme *Director of Educational License* (il n'est plus aujourd'hui impliqué dans Creative Commons mais continue à

travailler dans le domaine des Open Educational Ressources, OER). En 1999, Michael Stutz publia une nouvelle licence, la Design Science License, qui se positionnait discrètement comme une licence *copyleft* générale adaptée à tout type d'œuvres. Elle préfigurait ce que seraient ensuite les licences libres non logicielles (même si les auteurs avaient jugé utile d'y intégrer quelques spécificités liées au changement d'environnement : chaque nouvelle version doit par exemple porter un nouveau nom, ce qui favorise la création à partir d'œuvres existantes, mais complexifie la création d'œuvres collaboratives).

La culture Libre. La « fibre du Libre » est très ancienne dans la musique¹. Les artistes se sont très rapidement approprié l'usage d'Internet pour diffuser leurs œuvres, les faire connaître et en créer de nouvelles grâce aux contributions venant de toute part. Un nombre important de licences écrites pour la musique s'inspirèrent de la GNU GPL : la Free Music Public License (FMPL) et la Choral Public Domain Library License (CPDLL) en 1998, la Licence Art Libre (LAL) et l'Ethymonics Free Music License (EFML) en 2000, les Open Music Licenses (OML) (Green, Yellow, Red et Rainbow License) de l'Open Music Project lancé en 2001, l'EFF Open Audio License en 2001, les onze licences Creative Commons en 2002 (la Music Sharing License, aussi appelée CC By-NC-ND, étant la plus utilisée), la Loca Public License en 2002, l'Open Source Music License (OSML) en 2003 ou encore la licence IANG Ain't No GNU ! (IANG) en 2004. Les Open Music Licenses étaient certainement les plus complexes, préfigurant aussi ce que seraient les Creative Commons puisqu'elles étaient rédigées sur un socle commun et permettaient d'ajouter plusieurs interdictions, dont l'usage commercial et l'interdiction de la modification – et d'ailleurs Laurence Lessig² y contribua avant de

1. De nombreux exemples pourraient être cités, mais le mouvement Death Pact International est peut être le plus illustre : même s'il n'y avait pas de licence à proprement parler, il s'agit d'un concept de projet musical créé par un groupe anglais, Grey Wolves, au début des années 1980. L'idée était que toute œuvre distribuée sous le nom de DPI était librement utilisable, modifiable et redistribuable dès lors que cela était fait sous le même nom du Death Pact International (il y avait donc cette idée de fond commun, mais sans la reconnaissance de paternité maintenue par le jeu des licences).

2. À l'initiative du projet *Copyright's Commons* en 1999 au sein duquel il promut l'usage du signe *counter-copyright* [cc] pour signaler le partage de ses œuvres. Voir <http://cyber.law.harvard.edu/cc>.

se consacrer à Creative Commons¹. Ram Samudrala (artiste à l'origine de la FMPL) et Antoine Moreau (artiste initiateur de la LAL) furent ainsi les principaux penseurs de ce mouvement. Aujourd'hui, à l'exception de la Licence Art Libre (en version 1.3), la plupart de ces licences ont été abandonnées, ou retirées au profit des licences Creative Commons. Notons que les licences Creative Commons ne constituent pas l'unique objet de la fondation et qu'un certain nombre de projets dépasse le seul domaine des licences : ainsi le projet Public Domain Dedication ou la Certification de « mise dans le domaine public » furent imaginés dès 2002 (aujourd'hui abandonnés au profit de la licence CC-0 publié en 2009 et du projet Public Domain Mark de 2010). Un autre projet nommé Founders' Copyright avait été lancé dès 2002 afin de permettre aux créations d'une personne ou d'une entreprise – O'Reilly Associates ayant été moteur sur cet aspect – de tomber dans le domaine public passée la durée originale de 14 ans.

Les licences pour Wiki. À l'instar du droit d'auteur classique, les licences libres consacrent une place très importante à chaque auteur et contributeur, souvent sous la forme d'une mise à jour continue (sur le site et/ou dans les sources) d'une liste des différents contributeurs. Fluidifiant à l'extrême les processus de contribution et de collaboration, les projets formalisés au travers de wikis donnent néanmoins une place nouvelle et importante au projet lui-même (celui-ci étant très souvent cité en remplacement des auteurs réellement impliqués dans le contenu réutilisé). Cet usage, compréhensible pour des motivations de commodité, mais qui exprime aussi des changements culturels (le particulier s'éclipsant d'une certaine manière au profit du général) dut aussi être pris en compte dans la rédaction des licences (notamment en termes de formalismes s'imposant lors de la réutilisation du contenu). C'est cette raison qui suscita la mise à jour des licences Creative Commons au profit d'une version 2.5 (abandonnant de ce fait le projet d'une CC-wiki dédié) afin de préciser que la mention de la paternité devait être effectuée au profit du titulaire de droit ou d'une entité qu'il aurait désignée. Notons aussi que la FSF avait deux projets, aujourd'hui abandonnés, qui auraient pu permettre un tel usage : la GNU Simpler Free Documentation License (SFDL) ou la GNU Wiki (auquel renvoyaient les anciens brouillons de la GNU GFDL 1.3).

1. Lawrence LESSIG, *Code and Other Laws of Cyberspace*, New York : Basic Books, 2000.

L'Open Gaming. L'Open Gaming license est associée en 2000 sur le jeu *Dungeons & Dragon* par Wizards of the Coast, introduisant la notion de Document de Référence du Système (DRS) établissant leur jeu de rôle comme standard (en l'espèce le *d20 system*). La licence distingue les Open Game Content (OGC) des Product Identity (PI) : seuls les premiers (mécanismes du jeu) sont soumis à la licence *copyleft* alors que les seconds (tout ce qui est lié à la marque, aux figurines, histoires, graphismes, etc.) restent l'entière propriété de leurs auteurs. La portée de cette licence est donc à différencier de la *Licence ludique générale*, elle aussi publiée en 2000, qui portait la GNU GPL dans le domaine des jeux vidéos (en ajoutant quelques spécificités comme la notion de « noyau du jeu » définissant l'étendue de la licence d'après la volonté de l'auteur).

L'Open Access ou le libre accès. Le mouvement de l'Open Access trouve ses racines dans le milieu du vingtième siècle, mais c'est dans les années 1980, avec l'arrivée d'Internet et de l'informatique, qu'il put réellement se diffuser : perçu comme l'évolution logique des pratiques scientifiques à l'ère du numérique (favorisant les échanges, évitant de dupliquer des recherches et des erreurs, offrant une vision plus large de l'état de la science, etc.). L'Open Access repose sur un certain nombre de normes communes : le Protocole OAI-PMH élaboré en 1999 par l'Open Archive Initiative afin de favoriser l'interopérabilité entre les archives ouvertes ; les déclarations de Budapest (Budapest Open Access Initiative) en février 2002, Bethesda (Bethesda Statement on Open Access Publishing) en juin 2003 et Berlin (Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities) en octobre 2003. Ce concept englobe les publications électroniques en libre accès ou accès ouvert, c'est-à-dire qui offrent à tout utilisateur un droit d'accès immédiat, gratuit, irrévocable et universel des articles scientifiques en entier. Dans le mouvement d'Open Access, les auteurs autorisent un usage gratuit et sans restriction de leurs productions et se tournent ainsi naturellement vers les licences de type Creative Commons sans que des licences spécifiques n'aient été rédigées ni même ne soient recommandées. Par ailleurs, le principe repose sur la conservation par les auteurs de leur titularité lors de la publication, ce qui induit de changer les pratiques actuelles (le projet Sciences Commons recommande à cet effet l'ajout de certaines clauses afin d'atténuer les cessations de droits, au travers du Scholar's Copyright Addendum Engine).

L'Open Data. Les prémisses de l'Open Data sont plus lointaines qu'on ne l'imagine et il est ainsi possible de citer des projets comme l'Open Directory Project (ODP, base de données collaborative de sites web plus connue sous le nom de *dmoz1*) libéré par Netscape sous Open Directory License en 1999 (le projet est sous CC By-SA depuis juin 2011), Tela Botanica (base de données botanique) ou Open Street Map (base de données cartographique), respectivement lancées en 1999 et 2004 sous licence CC By-SA. Il apparut néanmoins quelques années plus tard que ces licences n'étaient pas nécessairement adaptées aux bases de données et que les projets nécessitaient de faire reposer leur efficacité sur une assiette plus large que le seul droit d'auteur, en incluant également le droit des contrats et le droit *sui generis* des bases de données. Au travers du projet Open Data Commons, l'OpenKnowledge Foundation fit ainsi rédiger spécialement des licences adaptées à ce besoin : la Public Domain Dedication and License (PDDL) – « Public Domain for data/databases » ; l'Attribution License (ODC-By) – « *Attribution for data/databases* » ; et l'Open Database License (ODC-ODbL) – « *Attribution Share-Alike for data/databases* ». Seule licence *copyleft* adaptée aux données, l'ODbL impose que toute base de données – modifiée ou non – soit distribuée sous la même licence. La première version « brouillon » (0.9) de la licence fut soumise à commentaires le 27 février 2009 (et publiée quelques mois plus tard). De nombreuses licences spécifiques existent aujourd'hui (telle la Licence Ouverte publiée en 2011 par la Commission Etalab en France pour les données de l'administration).

Police Libre. Les protections sur les polices de caractères ont très tôt été affirmées et il semblerait que leurs concepteurs puissent prétendre à des droits d'auteur lorsque toutes les conditions sont réunies, les polices étant assimilées à des logiciels. Quelle que soit la validité de ces droits, la présence de ces protections était susceptible d'empêcher l'utilisation de polices les plus répandues par les logiciels libres n'ayant pas souscrit les licences nécessaires. Le projet Free UCS Outline Font est certainement le premier à avoir diffusé des polices de caractères sous licences libres (GNU FreeFont). Ces polices sont sous GNU GPL avec la *font exception* qui empêche l'extension de la licence à tout document utilisant ces polices. L'association SIL (Summer Institute of Linguistics Inc.) International, aussi à l'origine de nombreuses polices libres (publiées pour répondre à ses ob-

jectifs d'étude et de développement des langages), est à l'origine de la licence SIL Open Font License (OFL). Anticipant tout frein à l'adoption de leur logiciel, de nombreuses industries ont ensuite diffusé des polices libres : les polices de caractères *Liberation font* par la société Red Hat en 2007, en version Sans, Serif et Mono (Oracle a contribué en 2010 la police *Liberation Sans Narrow* sous une licence légèrement modifiée) ; Google a aussi publié trois polices que sont *Arimo*, *Tinos* et *Cousine*, distribuées sous OFL, et deux autres polices destinées à Android sous licence Apache v2.

Open Hardware. Les produits de l'Open Hardware étant plus complexes, ils sont couverts par de multiples droits d'importance différente : les droits de propriété industrielle (brevets, dessins et modèles, topographies de semi-conducteurs) sont ici prédominants alors que le droit d'auteur est encore relativement peu utilisé (sauf pour ce qui concerne les logiciels embarqués, ainsi que la partie reprogrammable de certains composants). L'Open Source Hardware and Design Alliance a déjà joué un rôle structurant en proposant dès 2009 un label Open Source Hardware (fondé sur une définition dérivée de la FSD) ainsi qu'une nouvelle terminologie proposée en 2001 en remplacement de Free/Libre Open Source Hardware : *reables*. En pratique, chaque projet a adopté une politique différente en matière de licences : GNU GPL pour *FreeCores*, GNU LGPL pour le projet Opencores (1999), licences *copyleft* ou permissives pour le Linux-Fund.org (1999 – initialement Open Hardware Foundation) et l'Open Graphics Project (quelques licences propriétaires également pour ce dernier), ou encore la rédaction de licences spécifiques comme pour le projet Balloon ou Tucson Amateur Packet Radio (TAPR). Le projet Qi hardware, qui succède au projet OpenMoko, a pour ambition de créer un pot commun de matériels libres et recommande aujourd'hui la GNU GPL v3 (à la place de son ancienne politique par laquelle les plans étaient sous CC By-SA, le code sous GNU GPL v2 et les brevets sous une licence particulière). Le CERN a publié la version 1.1 de sa licence OHL (v1 en mars 2011 ; v1.1 en juillet 2011), qui encadre les plans et documentations d'équipements aux spécifications libres tout en imposant certaines obligations liées au partage des modifications des plans et à l'avertissement de toute mise en œuvre de ceux-ci. Dans un domaine aussi très technique et couvert par les brevets, citons le *BiOS Material Transfer Agreement*, outil juridique créé

en 2006 pour une plate-forme de partage encourageant la coopération et le développement de technologies (les brevets n'étant partagés qu'au sein de la communauté).

2. La consolidation et l'harmonisation des outils juridiques (depuis 2007)

Un besoin de mise à jour. Près de deux décennies après le lancement du mouvement du logiciel libre, la nécessité de moderniser et d'harmoniser les instruments juridiques s'est imposée. En effet, un certain nombre d'éléments ayant changé, pris en compte par de nouvelles licences ou non (le type de contributeur ou d'utilisateur, les utilisations et la « consommation » de l'informatique, le contexte législatif, etc.) et certaines utilisations ayant dépassé les objectifs initiaux des licences (surtout les plus utilisées, qu'il s'agisse des licences GNU ou des Creative Commons¹), il était temps de rééquilibrer l'équation à laquelle souhaitent répondre les licences libres et de prendre un peu de hauteur pour que toutes les initiatives opèrent de façon plus concertée. En cela, la multiplication des licences peut être vue positivement puisqu'elle permet globalement l'amélioration de certaines clauses (en termes de cession, de formalisme, etc.) ainsi que l'identification de besoins non nécessairement anticipés dans les licences initiales. À l'inverse, elle fut aussi vécue comme un frein, empêchant la combinaison de logiciels complémentaires sous licences libres.

Deux phénomènes suivirent cette première phase : une prise en main des licences au travers d'une pratique de plus en plus perfectionnée des utilisateurs (2.1.) ; une harmonisation des licences libres au sein de l'écosystème par les rédacteurs des licences (2.2).

2.1 Une évolution des pratiques entourant les licences

Le besoin de se recentrer sur les effets de la licence. Pour la majorité, les organisations qui rédigèrent une licence le firent initialement pour des besoins propres : s'assurer de l'accessibilité du code, réserver un contrôle

1. Voir Florian CRAMER, « The Creative Common Misunderstanding », dans : *FLOSS+ Art*, sous la dir. d'Améric MANSOUX et marloes DEVALK, OpenMute, 2008.

(et donc un modèle économique), encourager le développement de *plug-ins* complémentaires, nuire à un concurrent, etc. Inversement, la grande majorité des créateurs a pendant longtemps utilisé les licences libres sans véritablement lire, analyser, ni même comprendre les textes, se contentant au contraire de l'explication et de la philosophie qui s'en dégageait. Cela explique certainement que des licences très bien rédigées telles que l'OSL ne connurent qu'un succès mitigé alors que les licences GNU furent utilisées bien au-delà du cadre pour lequel elles avaient initialement été rédigées. Mais, passé l'engouement initial (ayant entraîné des utilisations contestables), il y eut un mouvement progressif de reprise en main par les utilisateurs. Le Libre a en effet eu pour conséquence de rendre l'utilisateur « acteur » et responsable du cadre dans lequel était utilisée la création.

Cela se traduit parallèlement par des situations nécessitant le changement de licences au sein de projets communautaires (2.1.1.) et l'établissement de pratiques permettant de modifier les licences utilisées (2.1.2.).

2.1.1 Les pratiques des utilisateurs : les changements de licences

Une nécessaire titularité pour le changement de licence. Afin de maintenir la confiance assurée par la licence et pouvoir valablement sanctionner des comportements déviants, un certain nombre de projets communautaires durent changer leurs licences. Cela pose rapidement une difficulté, car la plupart d'entre eux n'avaient pas anticipé une telle hypothèse lors du choix initial de la licence et n'avaient donc pas la capacité suffisante pour opérer une telle action. Pour rappel, les licences ne sont que des contrats de cession non exclusive de droits de propriété intellectuelle¹ et seuls les titulaires de droits (généralement les auteurs, sauf s'ils ont accepté de céder leur droit de façon exclusive ou consenti à déléguer cette opération) peuvent convenir d'une diffusion dans d'autres termes. C'est pour cette raison que de nombreux organismes, dont la FSF ou la SFC, demandent une cession pure et simple des droits relatifs aux projets qu'ils hébergent. Une telle pratique n'était néanmoins pas du goût de tous

1. « Les licences libres sont les licences par lesquelles l'auteur autorise la copie, la modification et la diffusion de l'œuvre modifiée ou non, de façon concurrente, sans transférer les droits d'auteur qui y sont attachés et sans que l'utilisateur ne puisse réduire ces libertés tant à l'égard de l'œuvre originale que de ses dérivés. Les œuvres ainsi diffusées sont qualifiées de libres. », CLÉMENT-FONTAINE, *loc. cit.*

(la création en 1991 du projet Lucid Emacs, *fork* de l'outil Emacs, a notamment été imputée au refus d'une cession systématiquement des droits sur le projet) et est restée longtemps mal comprise. De ce fait, quelques projets connurent des difficultés à changer leur licence, qu'il s'agisse de : documentations (l'exemple de Debian et de Fedora), logiciels (l'exemple de Mozilla, Squeak et du noyau Linux), « sites de collaboration multiauteurs » (Wikipédia) ou encore de bases de données (OpenStreetMap).

En matière de documentation : l'exemple de Debian (OPL à GNU GPL et MIT). La communauté Debian, notamment au travers des membres du groupe *debian-legal*, est particulièrement sensible aux enjeux de licences (systématiquement scrutées à l'aune du DFSG). Initialement rédigé à destination des logiciels, ce document n'a été appliqué que plus tardivement à tout contenu contribué (programme, documentation, image, son, etc.). Ainsi, en sa qualité de précurseur sur le sujet, la communauté avait naturellement mis le contenu de la documentation sous l'Open Publication License v1 lors de sa sortie en 1999. Quelques années plus tard, alors que d'autres licences fidèles au principe des quatre libertés avaient été publiées, la question se posa à nouveau avec d'autant plus d'acuité et de recul. Dès 2003, certains critiquèrent l'incompatibilité de l'OPL avec la GNU GPL, allant jusqu'à constater le non-respect de la DFSG le 14 mars 2004 (en raison d'un formalisme trop contraignant et susceptible de limiter les modifications, ainsi que d'une non-compatibilité vis-à-vis du « Dissident Test »). Depuis lors, et après plusieurs discussions animées sur les listes dédiées (notamment en 2006 et 2011), il fut choisi le 25 janvier 2012 de diffuser le contenu de la documentation sous GNU GPL (ou licence compatible) et licence Expat. Depuis cette date, toute nouvelle contribution peut être redistribuée et modifiée selon les termes de la licence Expat ou de la GNU GPL (version 2 ou ultérieure)¹. Près de 10 ans de procédure !

En matière de documentation : l'exemple inverse de Fedora (OPL à CC By-SA). Parallèlement, la communauté de Fedora avait entamé la même démarche de relicenciement du contenu sous OPL en 2009 au profit de la CC By-SA (conformément aux recommandations de David A.

1. Le contenu antérieur restant disponible sous OPL v1 dans l'attente de l'obtention d'un accord de tous les contributeurs (ceux-ci ont été contactés et la plupart ont accepté ce changement, la dernière phase de « consolidation » est programmée pour 2013).

Wiley). L'objectif étant de rejoindre les pratiques des autres communautés afin de favoriser le transfert de contenu (« *We can move from a tiny archipelago to the largest continent of open content under the flag of the CC BY SA* »¹). Cela fut néanmoins beaucoup plus rapide que pour Debian car Fedora possède une grande souplesse grâce au Contributors License Agreement imposé par Red Hat pour toute contribution (Debian s'étant tardivement rendu compte qu'en l'absence de cession à SPI, cette dernière n'était pas titulaire des droits comme la plupart l'avaient pensé). Cela permit d'opérer le changement en pas moins de trois années.

En matière de logiciel : l'exemple de Mozilla (MPL à la multilicence MPL/GPL/LGPL). Le navigateur Mozilla a aussi connu plusieurs changements de licence : lors de sa sortie en 1998 pour passer de la NPL à la MPL, puis en 2004 pour ajouter deux licences (la GNU GPL v2 et la GNU LGPL v2.1) au projet alors disponible à triple licence² à l'instar d'autres projets de Mozilla (tels que NSPR, NSS, PSM ou le moteur JavaScript). Ce relicenciement fut néanmoins relativement long puisque l'idée d'ajouter les deux licences GNU germa en août 2000 afin de conserver une licence adaptée au projet tout en rassurant les communautés du Libre (grâce à la compatibilité ainsi offerte). La formalisation du processus fut publiée le 19 septembre 2001 et l'achèvement de la procédure en janvier 2004. Même si la fondation s'était vu céder les droits que possédait Netscape en vertu de l'article 3 de la NPL (Alternative Licensing, permettant de choisir toute autre licence pour la redistribution du code), il était nécessaire d'obtenir l'accord de toutes les personnes ayant contribué au projet avant le 19 septembre 2001 (les contributions postérieures devant être faites sous triple licence). Ce fut un processus relativement long par lequel près de 450 contributeurs, personnes physiques ou morales, acceptèrent d'associer ces deux licences supplémentaires à leur code. Une fois ces autorisations récoltées, un script a effectué la mise à jour des en-têtes sur les fichiers affectés (plus précisément durant les cycles des versions 1.7a et 1.7b). Il fallut donc trois années pour corriger le statut de contributions apportées durant seulement deux années. Inversement, l'abandon

1. Propos rapportés par David A. WHEELER, « Make Your Open Source Software GPL-Compatible. Or Else. », 2002 (révisé en 2003), <http://www.dwheeler.com/essays/gpl-compatible.html>.

2. Une nouvelle version de la MPL a été publiée le 3 janvier 2012 et remplace dorénavant toutes les autres licences (elle est expressément compatible avec les licences GNU).

des licences GNU GPL et GNU LGPL lors de la diffusion sous MPL v2 (expressément compatible avec ces mêmes licences) fut immédiat.

En matière de logiciel : l'exemple de Squeak (APSL à MIT). Le logiciel Squeak était un projet collaboratif initialement diffusé par Apple sous licence Squeak. Devant la mise en cause de certaines des clauses de cette licence, la société a accepté de diffuser son code sous licence Apple Public Source License en 2006, puis licence Apache version 2 quelques mois plus tard. Cette cession ne concernait néanmoins pas les contributions individuelles versées sous la licence initiale du projet entre 1996 et 2006. Ainsi, avec l'aide de la Software Freedom Conservancy et la diffusion *a posteriori* d'un accord de relicencierement, une procédure relativement longue a permis de rendre le code disponible intégralement sous licence MIT (avec des portions sous licence Apache) depuis la version 3.10.2 sortie en mars 2010 (la procédure dura ainsi quatre années). Un exemple d'autant plus intéressant que le changement avait pour objectif, atteint, de permettre l'inclusion du logiciel au sein du projet Etoys du programme One Laptop Per Child, ce qui démontre une fois encore la diversité des stratégies ou leviers qui peuvent susciter le choix de certaines licences (et, en l'espèce, l'abandon d'une licence *ad hoc* au profit d'une licence plus utilisée et compatible avec la GNU GPL).

En matière de logiciel : l'hypothèse du noyau Linux (GNU GPLv2 à GNU GPLv3). À titre d'illustration, notons que la question du changement de licence (de la GNU GPL v2 seulement à GNU GPL v3) s'était aussi posée en 2007 lors de la publication de la GNU GPL v3. En effet, Linus Torvald étant clairement opposé à ce nouveau texte et refusant de revenir sur la limitation « *GNU GPL v2 only* » appliquée par lui et un certain nombre des contributeurs au noyau, empêchait Linux d'être distribué sous la troisième version de la GNU GPL, sauf à ce que leurs contributions soient réécrites (ou qu'ils changent d'avis et acceptent ce passage). La communauté de Linux et la renommée de Linus Torvald étant relativement fortes, cette position fut relativement bien suivie. Il fut néanmoins précisé que, compte tenu de la communauté dense et active du noyau, le changement serait tout à fait possible ultérieurement moyennant une substitution de tous les codes pour lesquels le changement de licence ne pourrait pas être obtenu.

En matière de « sites de collaboration multiauteurs » : l'exemple de Wikipédia. Lancée le 15 janvier 2001, l'encyclopédie en ligne Wikipédia utilisa la licence GNU FDL 1.2 sur son contenu durant de nombreuses années. Ainsi, tous les articles pouvaient librement être copiés, modifiés et distribués, ce qui permit à l'encyclopédie collaborative de grossir continuellement d'année en année, jusqu'à devenir une source de référence et une base de connaissance incontournable. Néanmoins, ce succès eut pour conséquence la professionnalisation de la gestion de l'encyclopédie ainsi que des usages qui en étaient faits. Dans ce cadre, le respect de la licence devint une condition indispensable au maintien de la confiance nécessaire aux contributions des milliers d'internautes. Or, le contexte dans lequel cette licence avait été rédigée justifia l'introduction d'une série de contraintes de formalisme lourdes et peu adaptées à l'usage effectif de Wikipédia : la GNU FDL imposait notamment 1) que la licence soit systématiquement recopiée au côté de la création, voire imprimée selon le nombre d'exemplaires tirés et 2) un *copyleft* très fort (calqué sur celui de la GNU GPL) et une série de stipulations complexes (avec notamment des sections invariantes ou encore des obligations différentes selon le format et le nombre de réutilisations du contenu). Le premier point posait certainement quelques difficultés pratiques lors de la reprise d'articles (le texte de la licence étant généralement plus long que l'article lui-même), mais le second emportait des effets juridiques (l'extension de la licence à toute œuvre contenant) non adaptés à l'effet souhaité de la licence.

C'est ce constat d'inadéquation qui motiva le changement de licence de Wikipédia en mai 2009. Ces conditions, dont la violation aurait normalement dû entraîner la résolution de la licence, étaient ainsi très rarement respectées et diminuaient d'autant la capacité des auteurs à faire respecter d'autres points de leur licence. Ainsi, l'ajout d'une Creative Commons CC By-SA 3.0 (plus claire, avec une étendue plus raisonnable et qui impose seulement des mentions légales succinctes ainsi qu'un lien vers la licence) eut pour effet de sécuriser la réutilisation du contenu de Wikipédia. Un tel changement aurait dû passer, à l'instar des autres projets, par une période transitoire de relicencement permettant à chaque contributeur d'accepter ces nouveaux termes. Mais deux écueils ajoutèrent à la difficulté de la démarche : le nombre inégalable de contributeurs à l'en-

cyclopédie (à l'époque à peu près 80 000 contributeurs par mois¹) et la simplification des modalités de contributions (beaucoup de contributions étant par exemple identifiables par la seule adresse IP – temporaire – du contributeur).

La solution différente et originale fut trouvée par l'ajout *a posteriori* d'une compatibilité *ad hoc* au sein de la GNU FDL utilisée. En effet, après avoir envisagé une licence adaptée au Wiki, la FSF accepta, malgré le profond désaccord de Stallman², l'idée de rédiger une nouvelle version transitoire de sa GNU FDL. Publiée le 3 novembre 2008, cette version 1.3 permettait une compatibilité expresse et limitée au profit de la CC-By-SA 3.0, par ailleurs spécialement limitée au seul bénéfice des « sites de collaboration multiauteurs » et pour tout changement opéré avant le 1^{er} août 2009. C'est sur cette base qu'a été modifiée la licence de Wikipédia, sans se passer néanmoins d'une procédure de validation par ses utilisateurs (étant juridiquement insuffisante – il aurait fallu une unanimité pour songer à un autre statut – celle-ci était néanmoins nécessaire d'un point de vue de gouvernance et de communication, pour intégrer la communauté dans ce processus). Aujourd'hui, le contenu est globalement disponible sous CC By-SA et, pour partie, cumulativement sous GNU FDL (un changement de licence qui put se faire, avec l'aide de la FSF, en moins de deux années).

En matière de bases de données : l'exemple d'Open Street Map (de la CC By-SA vers l'ODbL). En 2004, lorsque le projet OpenStreetMap fut lancé, le choix de la licence se porta naturellement sur la variante *copyleft* des licences Creative Commons (la CC By-SA) alors très en vogue. Cependant cette licence, essentiellement basée sur le droit d'auteur, montra rapidement ses limites dans une application à l'égard de données et/ou de base de données, la rendant difficile, voire impossible à mettre en œuvre selon les contextes : un flou juridique nuisible pour la réutilisation des bases de données (notamment par les acteurs économiques) et pour la communauté (puisqu'il faisait courir le risque d'une démotivation consécutive à l'utilisation de ces failles au profit d'un acteur ou projet tiers à la communauté). Il fallut donc s'assurer que la condition essentielle de

1. Voir <http://stats.wikimedia.org/EN/Tables/WikipediansEditsGt5.htm>.

2. Voir Lawrence LESSIG, « Enormously Important News From the Free Software Foundation », dans : *Blog de L. Lessig* (2008).

la licence, son *copyleft*, impose que toute base de données, modifiée ou non, soit distribuée sous la même licence. Cette prise de conscience remonte aux années 2008/2009 et conduisit à la mise en place d'un processus de changement de licence au profit de l'ODbL, soumis à un vote des membres de la fondation OSM (une décision favorable au changement de licence fut actée 26 décembre 2009).

Une procédure de transition en quatre phases fut alors établie par un *License working group* : la phase 1 (10 mai 2010) consista en l'association d'une double licence, CC By-SA 2.0 et OdbL 1.0, aux nouvelles contributions ; la phase 2 (10 août 2010) fut celle de la mise en place d'une procédure permettant aux anciens contributeurs d'accepter la nouvelle licence pour leurs anciennes contributions ; la phase 3 (17 avril 2011) fut l'étape à partir de laquelle la connexion au site et l'utilisation de l'API de la plate-forme imposèrent l'acceptation des termes de l'ODbL ; et la phase 4 (19 juin 2011) consista en la suppression des comptes et des données des utilisateurs n'ayant pas accepté la nouvelle licence. Un robot de nettoyage et de migration vint à cette fin éliminer les données des contributeurs n'ayant pas accepté le changement de licence : ce qui représenta moins de 1% des données à l'échelle mondiale et environ dix fois moins sur le territoire français (une cinquantaine de contributeurs ayant refusé le changement de licence – d'autres communautés locales telles que la Pologne ou la Macédoine furent plus touchées). La dernière étape de cette migration au profit de la licence ODbL 1.0 fut franchie par le projet OpenStreetMap le 12 septembre 2012 (date de la première publication sous OdbL seulement du fichier *planet* contenant un instantané des données – il fallut donc moins de trois années). Ainsi, alors qu'avec l'ancienne licence toute fabrication d'une carte devait être diffusée sous la même licence (sans nécessairement partager les données utilisées pour constituer sa carte ni même encadrer la conception de bases de données dérivées), aujourd'hui les cartes peuvent être produites sans contrainte de licence dès lors que sont mises à disposition toutes les améliorations apportées aux données OSM lors de la fabrication de sa carte.

2.1.2 Les pratiques des utilisateurs : les ajouts aux licences

L'intérêt des ajouts aux licences. Parallèlement, nombre de projets choisirent d'utiliser des licences bien connues en y associant des termes

adaptés à leurs besoins spécifiques. Cela présente un intérêt pour le projet – bénéficiant d'une licence déjà fortement éprouvée et à la sécurité juridique plus forte – ainsi que vis-à-vis de sa communauté puisqu'on bénéficie ainsi de l'image (que l'on souhaite généralement bonne) de la licence ainsi que des efforts de vulgarisation et de formation déjà réalisés sur cette dernière. Sans surprise, car elles restent les licences les plus symboliques, ce sont donc les licences GNU qui bénéficièrent le plus de ces exceptions (la licence MPL initiale étant rédigée pour favoriser l'apparition de nouvelles licences, les ajouts au texte se firent au travers de ces dernières).

Interprétations et exceptions. Deux types d'ajouts sont généralement distingués : l'*interprétation* qui permet de préciser une clause imprécise ou équivoque afin de figer la portée de la licence tenant les parties et le juge ; l'*exception* qui modifie une licence préexistante en ajoutant une spécificité qui déroge aux termes initiaux et qui rend ainsi la licence finale plus ou moins contraignante¹. L'une ou l'autre trouve généralement sa place dans une clause jointe à la licence ou insérée dans les en-têtes des fichiers. Elles sont d'autant plus intéressantes qu'elles témoignent très souvent des effets recherchés par les utilisateurs des licences, divergeant parfois de ceux souhaités par les rédacteurs de ces licences. La majorité d'entre elles concerne la GNU GPL et avaient pour objectif d'adapter (et de réduire) ses effets à un certain type de logiciel, de langage ou encore de modèle économique.

La première interprétation de la GNU GPL. Lors de son utilisation de la licence GNU GPL, Linus Torvald ajouta une interprétation de la notion de *derived work* dans une précision apportée sur chaque en-tête de ses fichiers : « NOTE ! Ce copyright ne s'étend *pas* aux programmes utilisateurs qui utilisent les services du noyau par des appels systèmes normaux – cela est considéré comme une utilisation normale du noyau, et n'est donc *pas* soumis au régime des créations dérivées »². Une précision du même genre fut rédigée par Larry Wall lorsqu'il adopta la GNU GPL v2 dès sa publication en 1989 : « Pour ceux d'entre vous qui choisissez d'utiliser la GNU General Public License, mon interprétation de la

1. À noter que, dans la pratique de la FSF, la notion d'exception se limite à l'ajout de permissions additionnelles (et non de termes potentiellement plus contraignants).

2. Traduction libre, voir <https://www.kernel.org/pub/linux/kernel/COPYING>.

GNU GPL est qu'aucun script Perl tombe sous les termes de la GPL sauf mention explicite de votre part mettant ledit script sous les termes de la licence GPL »¹. L'une et l'autre sont célèbres et toujours d'actualité. En pratique, elles favorisent le développement d'applications accessoires ou dépendantes et sont considérées comme intimement liées au succès de ces logiciels. Cet usage semble néanmoins témoigner d'une époque révolue, la légitimité de ces auteurs égalait alors celle des organisations telles que la FSF et de telles interprétations étaient non seulement acceptées, mais aussi difficilement contestables. Dorénavant, ce type de précisions tendrait à prendre la forme d'exception (la doctrine s'étant en effet développée de telle sorte qu'il serait alors plus complexe d'expliquer les raisons d'une interprétation divergente plutôt que de simplement modifier la portée de la licence). L'interprétation sera donc limitée aux seules zones grises de la licence, le cas échéant, et dans l'absence d'interprétation publiée par le rédacteur de la licence lui-même.

Les exceptions associées aux « générateurs de codes ». Certains logiciels susceptibles d'intégrer leurs propres codes au sein des créations générées utilisèrent la technique des exceptions pour encadrer l'étendue de la licence afin que tout produit généré par ces outils puisse être utilisé conformément aux choix de leurs auteurs en termes de licence. Ainsi, certains fichiers du compilateur GCC (libgcc², libstdc++, libfortran, libgomp, libdecnumber, libgcov, etc.) se voient associer une exception *Runtime library exception* qui permet de produire des programmes sous d'autres licences que la seule GNU GPL (l'analyseur syntaxique GNU Bison s'est vu ajouté une exception similaire en 1996). À l'inverse, et cela est révélateur de l'aspect stratégique de la licence, le studio de développement EiffelStudio intègre des fichiers sous GNU GPL qui imposent aux utilisateurs l'utilisation de la même licence (sauf à ce qu'ils se tournent vers la licence commerciale).

La limitation de l'étendue de la GNU GPL au travers de la *GPL linking exception*. La *GPL linking exception* est aujourd'hui une exception « générique » réutilisée à l'identique, ou dans une forme similaire, par

1. Traduction libre, voir <http://dev.perl.org/licenses>.

2. Il s'agit peut être du premier composant pour lequel Stallman décida d'utiliser une exception : libgcc.c permettant d'implémenter des opérations primitives arithmétiques sur des machines qui n'avait pas d'instructions pour ces opérations, il considéra que cela ne devait pas emporter d'effet à l'encontre des programmes utilisateurs.

de nombreux autres projets. Ainsi, la *GPL linking exception* est utilisée sur des bibliothèques désirant une licence proche de la GNU LGPL en ce qu'elle permet expressément les liaisons entre programmes, mais sans requérir de conditions particulières, la GNU LGPL étant en effet relativement exigeante quant à la fourniture des informations nécessaires (même si non soumis au copyleft, certaines informations du code utilisant le composant sous GNU LGPL devront être fournies afin de permettre à l'utilisateur de modifier et tester des versions modifiées de ce dernier). Elle est actuellement utilisée par des projets comme GNU Guile, GNAT ou GNU Classpath – l'*Exception classpath* utilisée sur le projet GNU Glasspath (plus tard réutilisée sur Java par Sun) est sans doute la meilleure illustration de l'utilité de l'exception puisqu'elle permet de lier du code à la bibliothèque sans étendre la GNU GPL au produit final. Dans le même ordre d'idée, on peut évoquer la wxWindows Library License publiée en 1998 qui, même si elle porte un nom, consista en l'ajout d'une permission à la GNU Library General Public License pour autoriser la distribution sous forme propriétaire le code objet de logiciels ou bibliothèques basés sur la licence (s'inspirant en cela de la MPL). Plus généralement, les noyaux logiciels semblent majoritairement avoir été diffusés – à l'instar du noyau Linux – sous des licences permettant expressément l'utilisation ou l'interaction avec des applications : qu'il s'agisse de l'eCos License 2.0 qui remplaça la Red Hat eCos Public License (ou RHEPL, une licence dérivée de la MPL) pour limiter l'effet de la licence GNU GPL aux seuls fichiers d'origine ou la licence du système d'exploitation Real-Time Executive for Multiprocessor Systems (ou OS RTEMS) qui se compose de la GNU GPL (version 2 ou supérieure) avec une exception particulière inspirée de l'exception GNAT Modified General Public licence (permettant ainsi la liaison avec d'autres composants). Enfin, notons que la GNU LGPL est elle-même une exception à la GNU GPL depuis leur version 3.

La limitation de l'étendue de la GNU GPL au travers de la FOSS exception. La *FOSS exception* (qui recoupe aussi différentes variations) permet de lever les problématiques d'incompatibilité entre licences dans un contexte *open source* seulement. Les composants *open source*, pour autant que leur licence figure dans la liste annexée à cette exception, peuvent ainsi continuer à être régis par leur licence d'origine même s'ils sont combinés ou intégrés au logiciel sous licence GNU GPL. Son utilisation fut

banalisée par MySQL, la licence GNU GPL + *FOSS Exception* remplaçant en 2004 la LGPL auparavant utilisée. L'intérêt d'une telle démarche (poussée en interne par Zak Greant) est de maintenir le bénéfice d'une diffusion et réutilisation dans un mode *open source* tout en conservant les contraintes de la GNU GPL pour une utilisation en dehors de l'*open source* (MySQL reste utilisé par de nombreux programmes sous licences propriétaires/commerciale). Même si les idées initiales restent les mêmes, on retrouve de nombreuses variantes à cette exception, se distinguant généralement par le nombre et le type de licences compatibles : par exemple, DigiRule Solutions's Free and Open Source Software (FOSS) License Exception, Kyoto Products FOSS License Exception (qui liste 11 licences), Hamsterdb FOSS License Exceptions (qui liste 8 licences), Stanford University FOSS License Exception v1.0 (qui en liste 12), etc.

Les exceptions dédiées aux langages. De nombreux concepteurs de langage ont reproché à la GNU GPL d'être trop inspirée des mécanismes du langage C, un héritage qui empêcherait de prendre en compte les spécificités des autres langages. Certains langages adoptèrent ainsi leur propre licence (la licence Artistic pour Perl en 1991, la licence Ruby de 1995, la licence PHP en 1995 ou encore la licence Falcon en 2003), mais la plupart se tournèrent vers les licences les plus connues (GNU GPL en tête, mais aussi MPL, BSD, EPL, Affero GPL, etc.). Parmi ceux qui adoptèrent une licence GNU (AGPL, GPL ou LGPL), plusieurs ajoutèrent une exception : telle la L LGPL (Lisp Lesser General Public Licence) rédigée dès 2000 pour le langage Lisp, la GNU Classpath Exception utilisée pour Java (la Enhydra Public License fut rédigée avec les mêmes objectifs), la GNAT Modified General Public License pour Ada (faisant suite à la création de l'entreprise par les fondateurs du projet en 1994), etc.

L'ajout de licences aux licences. Dans la même idée, beaucoup d'utilisateurs de licence s'aperçurent qu'il serait plus radical, et parfois plus simple, d'ajouter une licence réputée aux côtés de leur licence spécifique. On parle alors de multilicence (une seule et même création est simultanément disponible sous différentes licences) ou « double licence » lorsque la licence libre n'est qu'une option au côté de licences propriétaires classiques. Une telle solution présente un certain nombre d'aspects négatifs et positifs. Parmi les désavantages, il convient de préciser : la complexité qu'un tel schéma peut revêtir pour un utilisateur non averti (il est cepen-

dant tout à fait possible de ne communiquer que sur l'une seule des licences et de n'évoquer les autres licences qu'en présence de personnes ou projets susceptibles d'y être attentifs); la nécessité de distribuer sous les mêmes licences toutes les contributions souhaitant être réutilisées dans le projet; le risque accru concernant les *forks* (division de la communauté en deux projets distincts) qui deviennent encore plus dangereux et contreproductifs puisque les différents projets sont susceptibles d'être diffusés sous des licences incompatibles (au détriment du seul projet d'origine ou des deux). La solution a aussi son lot d'avantages : une meilleure compatibilité *downstream* (c'est-à-dire permettant la réutilisation du code au sein d'autres projets), le bénéfice de la renommée de toutes les licences, une consolidation des cessions de droits et, surtout, le maintien d'une mécanique propre à la licence initiale tout en permettant à d'autres communautés de profiter de ces développements. Ainsi, le langage de programmation Falcon, initialement sous licence Falcon Programming Language License (une licence proche de la licence Apache v2 optimisée pour Falcon, le langage de programmation de scripts, modifiée en 2008 pour empêcher l'extension de la licence aux scripts) a ensuite été cumulativement diffusé sous GNU GPL version 2 dès la version 0.8.10; la licence Artistic a doublé la distribution du langage Perl sous licence GNU GPL; la MPL s'est vu adjoindre les GNU GPL et LGPL dans les années 2000.

L'action des Commercial Open Source Software (COSS). Ce terme est parfois utilisé pour désigner des projets édités par des éditeurs commerciaux qui cherchent à diffuser une version *open source* tout en valorisant de façon classique des fonctionnalités (logicielles ou non) plus évoluées pour ses clients (créant des *vendor lock-in* – des mécanismes rendant dépendant d'un éditeur – en tout point similaires à ceux des logiciels traditionnels). On observe que, depuis quelques années, le modèle de double licence poussé par MySQL, Talend, Trolltech, etc. (qui proposaient une licence commerciale alternativement à une licence *open source*) perd en intérêt. En effet, si les clients industriels « achètent » volontiers pendant quelques années la sécurité juridique par ce biais (agissant ainsi sur recommandation de leurs conseils juridiques), ils se tournent aujourd'hui de plus en plus vers les versions communautaires dont ils perçoivent mieux les avantages ainsi que les conditions. Néanmoins, les éditeurs perfectionnent et maximisent aussi les marges de manœuvre dont ils dis-

posent du fait des licences libres, s’approchant parfois dangereusement de la frontière entre libre et fermé (ces mêmes éditeurs associent par exemple à la plupart de leurs contrats – services, assistance et support, avec les distributeurs, etc. – des clauses par lesquelles les cocontractants s’empêchent de bénéficier ou promouvoir la version communautaire libre).

2.2 Une harmonisation par la modification et consolidation des licences

2.2.1 Une consolidation par la GNU GPL

Le rôle complexe de la GNU GPL. La GNU GPL joue un rôle compliqué dans la consolidation de l’écosystème des licences, alliant deux paradoxes. Le premier concerne le rôle (et la légitimité conséquente) de la GNU GPL : à la fois fondatrice et « leader historique » qui tend à réunir les autres licences sous sa coupelle et « extrémiste du libre » qui, par sa position souvent inconditionnelle, détourne parfois certains utilisateurs (raison pour laquelle Linus Torvald rejeta la dernière version de la GNU GPL). Le second paradoxe concerne l’étendue très forte de la GNU GPL qui, contrairement aux licences de type MPL, emportait l’extension de la GNU GPL sur des composants initialement soumis à d’autres licences (ce qui était relativement agressif ou contraignant pour les développeurs). Cette étendue forçait la compatibilité sans laisser de réelle marge de manœuvre aux rédacteurs de nouvelles licences (beaucoup de licences furent ainsi modifiées pour devenir compatibles avec la GNU GPL : la BSD en 1999, l’Eiffel Forum License v2 en 2008, etc.). Au fur et à mesure de son adoption, la licence s’est néanmoins détachée graduellement du contrôle de la seule FSF et le juriste Eben Moglen assista Richard Stallman dans la rédaction des licences qui suivirent la GNU GPL v2 (personnellement dans un premier temps et ensuite par l’intermédiaire du Software Freedom Law Center qu’il dirige). La troisième version de la GNU GPL vient alléger les deux paradoxes évoqués.

Une conservation de l’esprit initial. Il est ainsi intéressant d’écouter Eben Moglen expliquer qu’en dépit des modifications importantes que connut la GNU GPL, la dernière version conserve l’esprit initial qu’elle replace dans l’époque actuelle :

La GPLv3 transpose la constitution de RMS pour le logiciel libre dans le nouveau siècle. Elle rend les principes juridiques du logiciel libre véritablement mondiaux, non centré sur le seul droit américain. Elle a renforcé les défenses de logiciels libres contre les attaques en matière de brevets, juste à temps pour la guerre des brevets que nous avons depuis longtemps prédit en cas d'acceptation du brevet logiciel. Elle empêche aussi l'emprisonnement des logiciels libres dans des produits matériels qui ne permettent pas à leurs propriétaires de modifier le logiciel incorporé. La GPL v3 a anticipé les enjeux d'aujourd'hui et nous aidera à relever les défis de demain¹

En pratique, cette nouvelle version opère des changements considérables afin de maintenir l'objectif initial (la liberté des utilisateurs dans une société qui a évolué), tout au plus pourrait-on faire remarquer que ces évolutions dépassent parfois les attentes des utilisateurs de la licence, mais force est de constater que le travail fourni par la FSF répond à la mission dont elle a, plus ou moins implicitement, la charge : plus que de faire évoluer un texte juridique, elle a pour mission de protéger les libertés des utilisateurs.

Les problèmes posés par la faille des logiciels utilisés en tant que service (ASP loophole). Conçues à l'époque où les logiciels s'échangeaient encore principalement sur support magnétique, les premières licences libres associaient l'acte de distribution à celui de la sortie de la sphère privée (emportant les contraintes nécessaires pour assurer les libertés des utilisateurs subséquents). Néanmoins, avec le développement du réseau, le nombre de logiciels libres utilisé en « Software as a Service » (Saas), c'est-à-dire comme un service², devint de plus en plus important. En l'absence de distribution du code, de nombreux géants du web ont pu construire leurs services sur la base de composants de logiciels libres sans contribuer directement à leur propre modification, cassant la confiance de certains développeurs pour qui un tel usage aurait dû être couvert par la GNU GPL. Au début des années 2000, plusieurs nouvelles licences vinrent encadrer cet usage : l'Apple Public Source License (APSL) publiée en 1999, la Reciprocal Public License publiée en 2001 par Scott

1. Traduction libre des propos retranscrits par Josh Gay le 29 juin 2012, <http://www.fsf.org/blogs/licensing/gnu-gplv3-turns-5>.

2. Installés sur les serveurs d'une entreprise, ces logiciels permettent d'offrir des services à une multitude des clients de cette dernière sans être matériellement accessibles aux utilisateurs finaux.

Shattuck (de la société Technical Pursuit Inc.) et mise à jour (version 1.5) en 2007, l’Affero Public License publiée en mars 2002 par la société Affero avec l’aide de la FSF ou encore l’Open Software License en 2002 (d’autres suivirent et reprirent ensuite ces clauses, telles que la Common Public Attribution License (CPAL) en 2007 ou l’European Union Public Licence lors du passage en version 1.1 en 2009). Renonçant à son idée de l’intégrer directement au sein de sa nouvelle GPL (mais le projet figurait encore dans l’article 7b.4 de son second brouillon¹), la FSF conçut une nouvelle licence, la GNU Affero GPL, qui contient un article 13 obligeant le licencié qui utilise le logiciel pour fournir un service à des utilisateurs via le réseau à redistribuer les modifications qu’il aurait lui-même apportées à ce dernier.

L’encadrement de mesures techniques de protection et la Tivoïzation. Le terme Tivoïzation est un néologisme définissant la création d’un système qui inclut des logiciels libres, mais utilise le matériel électronique pour interdire aux utilisateurs d’y exécuter des versions modifiées ne disposant pas des signatures électroniques du constructeur (technique initialement mise en œuvre par TiVo sur ses enregistreurs vidéos numériques, mais ensuite réutilisés pas d’autres constructeurs tels que Tomtom pour les GPS). La GNU GPL v3 intègre et encadre ce dispositif en interdisant expressément de limiter l’usage d’un produit embarquant du code sous GNU GPL v.3 à la seule version fournie par le constructeur (sous peine de perdre le bénéfice de la licence). Cette clause est strictement réservée aux « produits de consommation » (donc non applicable à un usage interne ou entre professionnels), mais a un impact concret dans le domaine de l’embarqué (de nombreux constructeurs refusant de permettre une telle mise à jour par l’utilisateur). Torvalds s’est catégoriquement opposé à cette clause qui limite certaines utilisations du logiciel (alors qu’à

1. Prévue pour la version 3 de la GNU GPL, le rejet de la clause Affero (qui aurait eu un impact important et difficilement maîtrisable sur les logiciels utilisés en Saas puisque cet usage serait devenu « déclencheur » de la licence, entraînant l’obligation de mettre à disposition les sources ainsi que toute modification apportée au logiciel) entraîna de violentes accusations (certains n’hésitant pas à dire que « la GPL est la nouvelle BSD », voire à lui affubler l’appellation de « SHING GPL » en référence à « Sun HP IBM Nokia Google », principaux bénéficiaires de cette clause).

ses yeux la licence doit au contraire favoriser les usages) et refuse depuis lors de changer de licence au profit de la dernière version¹.

La résolution immédiate. Comme évoqué précédemment, la sanction initiale à toute violation de licence libre était initialement très radicale puisqu'elle entraînait la résolution pure et simple de la licence, rendant contrefacteur l'utilisateur quant à tous les usages ultérieurs. Utile pour faire respecter les licences, cette sanction était néanmoins parfois trop sévère et faisait courir un risque important au secteur industriel susceptible (comme tout autre²) de violer une licence par ignorance. Ainsi, un mécanisme d'atténuation à progressivement été introduit au sein des licences quant aux effets d'un tel non-respect, permettant au licencié de corriger son comportement, voire d'être automatiquement réintroduit dans ses droits, à certaines conditions. La FSF a intégré cette attente en organisant au sein de la GNU GPL v.3 un système de « rétablissement des droits » lorsqu'il s'agit d'une première violation et que celle-ci est rapidement corrigée. Par la suite, la FSF communiqua sur l'importance de ce changement face aux nouveaux usages industriels réalisés à partir du noyau Linux (Android en premier lieu) afin d'inciter le projet Linux à utiliser la dernière version de la licence³. Sans succès.

La recherche d'une meilleure sécurité juridique et l'enjeu en matière de brevets. Bénéfice sans doute le plus important de l'apport des licences industrielles, la recherche d'une meilleure sécurité juridique est commune à la majorité des licences qui se succédèrent dans les années 2000 : internationalisation des licences, meilleure définition de l'œuvre dérivée pour les licences *copyleft*, généralisation des licences en matière de brevets, stipulation relative à la loi compétente, encadrement des mesures techniques de protection, etc. L'accélération de la brevetabilité des logiciels (de façon explicite aux États-Unis et Japon, de façon masquée en Europe où les logiciels ne peuvent être breveté en tant que tels) fut certainement vu comme le plus grand danger des dernières années : beaucoup

1. Voir à cet égard Linus TORVALDS, « Re : Dual-Licensing Linux Kernel with GPL V2 and GPL V3 », message sur la liste de diffusion *Linux Kernel Mailing*, 13 juin 2007.

2. Citons à cet égard Richard Stallman qui, en 2011, admis que plusieurs versions de son logiciel GNU Emacs ne respectaient pas la GNU GPL (« *Emacs distributions are not GPL-compliant* ») <http://lists.gnu.org/archive/html/emacs-devel/2011-07/msg01155.html>. Il est toujours bon de se rappeler que même les meilleurs sont faillibles.

3. Voir Brett SMITH, « Android GPLv2 termination worries : one more reason to upgrade to GPLv3 », 18 août 2011, <http://www.fsf.org/news/android-termination-upgrade-gplv3>.

plus difficiles à contourner, voire même à identifier, ils laissent planer une menace constante qui nuit à la pérennité (et donc l'adoption) des logiciels libres. Les sociétés peuvent se protéger contre un tel risque en développant elles-mêmes des portefeuilles de brevets défensifs, mais une telle pratique n'est pas possible pour un projet communautaire (même s'il est possible de se positionner contre de tels brevets ainsi que l'a publiquement fait Mozilla en 2004) qui peut au mieux bénéficier de la protection (plus précaire) de grands groupes (il s'agit d'initiatives de regroupements de brevets à titre défensif – *patent pools* individuels¹ ou collectifs² – avec pour objectif d'immobiliser toute action tierce en contrefaçon de brevets logiciels : on les intitule *patent angels*, *anti-patent trolls* ou encore *reverse patent troll*). Ainsi, si la FSF s'est très tôt positionnée contre les brevets logiciels (dès 1990, intégré l'année suivante dans les GNU GPL et LGPL), ce n'est que dans les années 2000 que les clauses des licences relatives aux brevets furent améliorées de sorte à encadrer le risque afférent (pour les industries comme les communautés) et mettre un terme à la campagne de FUD (*Fear, uncertainty and doubt*) menée contre les logiciels libres par certains éditeurs. Un effort similaire fut donc prévu dans le cadre de la nouvelle GNU GPL, d'autant plus qu'un nouvel artifice fut révélé dans l'accord conclu entre Microsoft et Novell puisque, Novell ne pouvant bénéficier d'une licence sur les brevets de Microsoft sans partager ce bénéfice, une situation préférentielle fut organisée afin qu'elle puisse valoriser (commercialement et en termes de communication) la promesse faite par Microsoft de ne pas agir à l'encontre des clients de sa distribution Suse. L'article 11 de la GNU GPL v3 vient donc améliorer la portée de la licence en matière de brevets (relative à toute invention contenue dans la version contributive) et étendant ses effets aux bénéficiaires de promesses de non-agression (qui doivent alors choisir entre rendre le code source public, renoncer au bénéfice de la licence ou étendre celui-ci).

1. Plusieurs milliers de brevets sont ainsi regroupés en garantie par des sociétés comme IBM, Nokia, Sun, etc., notamment l'IBM Statement of Non-Assertion of Named Patents Against OSS, le Sun Patent Program ou encore le Novell Statement on Patents and Open Source Software.

2. Notamment l'Open Source Development Laboratory (OSDL), l'Electronic Frontier Foundation (EFF, avec notamment l'initiative Patent Busting Project) ou l'Open Invention Network (OIN) auquel ont souscrit des sociétés comme Sony, IBM, NEC, Red Hat, Philips et Novell. La Foundation for a Free Information Infrastructure (FFII) et l'initiative *no software patents* affirmant que la constitution de regroupements de brevets à titre défensif est une mauvaise solution par rapport à la lutte pour imposer la non-brevetabilité des logiciels.

Une prise en compte des nécessités d'une meilleure compatibilité. Si l'on en croit les études publiées par les sociétés éditrices d'outils d'audit de code, il y aurait plusieurs milliers de licences différentes actuellement utilisées. En pratique, il s'agit de la centaine de licences répertoriées par l'Open Source Initiative, auxquelles s'ajoutent les versions antérieures et les modifications apportées par le biais d'interprétation ou d'exception. La licence GNU GPL est en tête en termes de couverture de code, mais le résultat est plus équilibré, notamment au profit de licences telle que la licence Apache, lorsqu'on se focalise sur les logiciels les plus importants. Cette multiplication de licences destinées à libérer l'utilisateur du contrôle des éditeurs aboutit néanmoins à une difficulté inhérente à la technique utilisée (le contrat). D'autant plus paradoxale au regard des objectifs des différentes licences, la notion d'incompatibilité décrit la situation où les termes contradictoires de licences libres entraînent l'impossibilité de les respecter simultanément et, conséquemment, de combiner plusieurs composants libres (soit en raison de leurs spécificités, soit parce qu'elles sont elles-mêmes *copyleft*). À l'instar de l'*ASP loophole*, la problématique de l'incompatibilité a longtemps été laissée dans une zone grise. Des licences comme l'Apache License ou l'Artistic License étaient considérées comme compatibles avec la GNU GPL par leurs rédacteurs respectifs alors qu'elles n'avaient pas été rédigées comme telles (l'incompatibilité la plus symptomatique, et longtemps problématique, fut par exemple entre la licence GNU GPL version 2 et l'Apache version 2, cette dernière contenant une clause plus protectrice en matière de brevet qui n'aurait pas été respectée si le composant s'était retrouvé couvert « seulement » diffusé sous GNU GPL). La nécessité de trouver une issue à ces enjeux n'est finalement que du bon sens et les rédacteurs de licences se sont aussi emparés du sujet (les exceptions FOSS ou *linking* sus-citées avaient une intention similaire), considérant que « le point important n'est pas les différences, mais au contraire le but commun ¹. Aujourd'hui, la compatibilité est assurée au sein de plusieurs licences par différents mécanismes : l'existence d'une clause expresse (la GNU LGPL v2.1 permet, par exemple, une diffusion sous GNU GPL v2), la compatibilité conditionnée (telles la CeCILL ou l'EURL qui permettent une rediffusion sous GNU GPL en cas de

1. Tiré d'un courrier de Lawrence Lessig, disponible sur <http://creativecommons.org/weblog/entry/5709>.

combinaison avec des composants sous cette licence). Depuis sa dernière version, la GNU GPL v3 contient un article spécifique destiné à renforcer une compatibilité à son profit, c'est-à-dire la capacité à respecter un nombre accru de licence en diffusant sous GNU GPL v3. Parmi les autres licences remarquables quant à cet aspect, il est possible d'évoquer l'European Union Public License v1.1 compatible avec cinq licences libres *copyleft* (l'OSL v.2.1 et v.3.0, la CPL v1.0, l'EPL v.1.0, la CeCILL v2.0 et la GNU GPL v.2.0), la nouvelle version de la Mozilla Public License publiée en 2012 et rédigée pour être compatible avec la licence Apache, ainsi que les licences GNU (Affero, LGPL ou GPL). La seconde version de l'EURL a pour objectif principal d'étendre la liste des licences compatibles. De la même façon, Creative Commons travaille aussi sur une compatibilité avec des licences proches de la CC By-SA (notamment la Licence Art Libre ouverte à une telle compatibilité depuis sa version 1.3 à condition d'une réciprocité). Malheureusement, la notion de compatibilité elle-même diverge selon que la licence soit très utilisée (telle la GNU GPL ou les Creative Commons qui cherchent surtout à favoriser l'usage de leur licence par ce biais) ou plus spécifique (telles que l'EURL ou la CeCILL qui souhaitent surtout que le code puisse être repris par d'autres projets, quitte à ce que la licence soit changée au profit d'une autre compatible).

2.2.2 Un processus de rationalisation : abandon et sélection

Le rôle des organisations du secteur. Les principales organisations du secteur ont joué un rôle essentiel : que ce soit la FSF, le SFLC et la SFC en accompagnant l'usage de licence GNU ; l'OSI en entamant une campagne contre la prolifération des licences, mais aussi toutes les autres organisations qui, par leur structuration, ont favorisé la diffusion de « bonnes pratiques » (telles que les Fondations Apache ou Eclipse qui conditionnent l'adhésion au respect d'une certaine politique en matière de licence). Dans les années qui suivirent, d'autres entreprises ont aussi joué le jeu : Sun a retiré sa licence SISSL en 2005 et SPL en 2006, IBM fit de même pour la licence CPL en 2009 (ainsi que l'Intel Open Source License, la Jabber Open Source License, la MITRE Collaborative Virtual Workspace License).

Le travail mené au sein de la fondation Creative Commons. Alors que le projet comptait à son lancement onze licences différentes en 2002,

ce nombre a été réduit dès 2004 avec la publication d'une seconde version des licences *Creative Commons* utilisée sans l'option d'attribution, répondant d'une pierre deux coups au faible attrait constaté pour ces licences ainsi qu'aux difficultés rencontrées lors de l'internationalisation des licences puisque l'option ne pouvait être retenue dans les pays accordant une place aux droits moraux de l'auteur. Parallèlement, certainement plus par naïveté que calcul, Creative Commons a incité et parfois accompagné de nombreuses demandes en rédigeant un certain nombre de licences spécifiques au *sampling* fin 2003 (*Sampling*, *NonCommercial-Sampling+*, *Sampling+*), aux pays en voie de développement (*DevNations*, une *CC By* limitée aux pays en voie de développement) en septembre 2004, ainsi qu'un projet non abouti en matière d'éducation en 2003. Les licences *DevNations* et la *Sampling License* furent retirées en 2007, ne répondant pas aux libertés minimales de diffusion non commerciale souhaitée par le projet. Les autres furent abandonnées en 2011 (la *PDD* fut remplacée par la *CC-zero* ainsi que la *Public Domain Mark*) en raison de leur faible utilisation, d'une volonté de clarté et de la recherche d'une meilleure compatibilité entre les licences. La *Creative Archive Licence* a ainsi permis la publication, de 2005 à 2006, d'un certain nombre de créations en provenance d'établissements publics britanniques (*BBC*, *British Film Institute*, *Université Ouverte*, etc.) afin de permettre une réutilisation non commerciale particulièrement encadrée (ainsi, celle-ci était limitée au seul territoire du Royaume-Uni et interdisait tout usage promotionnel ou politique).

Le projet de lutte contre la prolifération lancé par l'OSI. Fin 2007, l'*Open Source Initiative* a lancé un nouveau projet (complémentaire à celui de labellisation) de lutte contre la prolifération des licences. À cette fin, les licences sont depuis cette date classées en plusieurs catégories : 1) les licences qui sont populaires et largement utilisées ou avec une large communauté ; 2) les licences répondant à des besoins spécifiques/particuliers ; 3) les licences qui sont redondantes avec les licences les plus populaires ; 4) les licences qui ne sont pas réutilisables ; 5) les licences autres/diverses. Ce projet répond à une demande formulée dès 2004 au sein des administrateurs de l'OSI et qui avait donné naissance à un rapport publié en 2006 (*Report of License Proliferation Committee and draft FAQ*). Ce travail a permis d'identifier neuf licences « populaires » (*Apache License 2.0* ; *BSD 3-Clause New or Revised License* ; *BSD 2-Clause Simplified or*

FreeBSD License ; GNU General Public License (GPL) ; GNU Library or *Lesser* General Public License (LGPL) ; MIT license ; Mozilla Public License 2.0 : Common Development and Distribution License ; Eclipse Public License). L'OSI a ainsi contacté de nombreuses organisations afin de procéder à cette réduction du nombre de licences actives. Cette nouvelle mission que s'est attribuée l'OSI, tout à fait louable, a néanmoins froissé un certain nombre de personnes puisque la sélection « arbitraire » du comité en charge avait ensuite un impact sur l'usage de la licence (ainsi Lawrence Rosen, alors qu'il n'était plus conseil juridique de l'OSI, s'était clairement manifesté contre la classification de ses deux licences dans les catégories des licences « redondantes » ou « autres »).

Une obsolescence de fait. L'évolution des usages et des technologies ne cesse de remettre en cause nos paradigmes. De ce fait, un nombre croissant des licences utilisées devient obsolète sans que leurs rédacteurs de l'époque n'aient de réel contrôle sur celles-ci (la licence Artistic est à cet égard un exemple éclairant puisqu'en l'absence de clause permettant la mise à jour de la licence, deux versions cohabitent sans que le passage de l'une à l'autre ne puisse se faire aisément¹). Une « sélection naturelle » a amené les utilisateurs à privilégier le choix de licences mieux connues et assurant une meilleure compatibilité avec les projets existants.

2.3 L'importance du perfectionnement des processus

L'introduction d'une édition collaborative des licences. La majorité des licences utilisées par des communautés furent discutées avec celles-ci avant d'être définitivement publiées². Ainsi Stallman partagea-t-il sa GNU GPL avant de la publier ou Netscape fit circuler des brouillons avant toute publication. De nombreuses autres tombèrent néanmoins dans

1. Notons à cet égard que la fondation Perl considère, sans nécessairement le justifier pleinement, la seconde version de l'Artistic comme étant compatible avec la première et que Larry Wall accepte volontiers de relicencier ses contributions sous Artistic v2 (mais cela ne couvre malheureusement que ses propres contributions).

2. Étonnamment, l'importance des communautés dès l'édiction des licences n'a pas été aussi naturelle que l'on pourrait le penser. Voir notamment, au sujet des licences CC Sampling, Victor STONE, *ccMixer : A Memoir OR How I Learned to Stop Worrying about the RIAA and Love the Unexpected Collaborations of Distributed Creativity During the First Four Years of Running ccMixer*, février 2009, URL : <http://fourstones.net/ccmixer-a-memoir>.

le premier travers du droit qui est d'en faire une discipline réservée (quitte à ensuite connaître plusieurs révisions successives pour répondre aux attentes de chacun). Cependant, l'implication des communautés tend à se développer de plus en plus tôt : passant d'une discussion du texte final à une discussion de brouillon et enfin une consultation ouverte. Au fur et à mesure que les projets gagnent en importance et en contributeurs, il devient de plus en plus difficile, ou pour le moins très critiquable, d'imposer tout changement sans consultation des auteurs/utilisateurs de ces licences. En effet, le rédacteur se trouve très souvent dans la situation très particulière où il dispose de la faculté de changer un contrat utilisé par d'autres, en totale contradiction avec l'effet relatif des contrats selon lequel la convention à force de loi entre les parties (dès lors qu'il ne porte pas atteinte aux bonnes mœurs ou à l'ordre public) et n'est donc opposable qu'aux seules parties au contrat (seule une personne ayant consenti à un contrat peut se le voir opposer). D'une rédaction individuelle, le processus est devenu collectif, renforçant d'autant la légitimité et la force de ces licences comme outils de gouvernance des communautés. Par ailleurs, notons que même pour les organisations ayant anticipé contractuellement une telle faculté (par une cession exclusive des droits des contributeurs à leur profit ou grâce à une clause de la licence prévoyant une mise à jour de la licence à la discrétion du rédacteur de la licence), elles n'en usent généralement qu'après consultation et accord de leurs communautés. Les processus les plus aboutis furent certainement ceux mis en place pour la troisième version de la GNU GPL. La Free Software Foundation initia en 2006 une consultation des auteurs et communautés afin de comprendre les enjeux auxquels devait répondre la GNU GPL et les attentes des utilisateurs (notamment au regard d'un premier brouillon fourni en janvier 2006). Grâce à des outils dédiés permettant de récupérer, discuter et consolider tous les commentaires, un certain nombre d'attentes furent prises en compte : concernant les évolutions législatives, techniques, pratiques et industrielles. Après quatre itérations (chacune donnant lieu à une version consolidée et revue par Richard Stallman, Eben Moglen et Richard Fontana), la version finale de la GNU GPL v3 fut publiée le 29 juin 2007 et présentée comme la licence conforme à l'esprit d'origine, mais formalisant 16 ans de changements et acceptant un certain nombre de compromis pour répondre aux attentes des utilisateurs industriels comme

communautaires (la clause Affero fut certainement la plus polémique). Ce procédé fut repris dans des conditions assez semblables lors de la mise à jour de la Mozilla Public License en 2011. Ce processus collaboratif est aujourd'hui poussé encore plus loin au travers d'une initiative nommée *GPL.next* par Richard Fontana sur GitHub ¹.

Le processus de certification des licences a lui aussi grandement évolué. Aujourd'hui, quatre principes fondamentaux permettent de vérifier l'adéquation avec l'Open Source Definition : identifier la catégorie parmi celles identifiées par le projet de *License Proliferation* et essayer de limiter la duplication de licences (voir *infra*) ; et assurer une procédure transparente et pérenne d'examen des licences (dans les 60 jours en moyenne) et fournir une traçabilité du processus d'examen. De la même façon, cette structuration renforce l'intérêt de cette certification. La requête – en principe, faite par le rédacteur de la licence, mais le respect de cette règle est relatif – doit identifier le type de soumission, vérifier que tous les critères sont réunis pour cette requête et être complète (sujet clair, texte de la licence, demande d'inscription au système de suivi et toute autre information utile). Le demandeur doit lui-même s'inscrire à la liste de diffusion correspondante (*license-review*). Il existe plusieurs types de soumission d'une licence : lors de la mise à jour d'une licence (il faut alors inclure un lien vers l'ancienne, la nouvelle, et les modifications apportées à la licence), pour l'ajout dans une nouvelle catégorie, pour retirer une version, pour l'approbation d'une version déjà existante (il faut indiquer la catégorie et les justificatifs) ou pour l'approbation d'une nouvelle licence – ou licence utilisée par une seule entité (il faut soumettre la préversion, justifier, comparer vis-à-vis des licences existantes, recommander une catégorie). Par la suite, la certification de la licence est discutée sur la liste dédiée au moins une trentaine de jours (l'outil de suivi de l'OSI fournissant le statut approprié de chaque tâche) et une synthèse des recommandations est présentée au bureau de l'OSI (avec copie à la liste) par la personne en charge de la labellisation des licences. Le bureau de l'OSI dispose alors du pouvoir de prendre une décision finale ou de demander des informations complémentaires. Pour finir, la personne en charge des

1. Service web d'hébergement et gestion de versions décentralisée permettant à chacun de créer sa propre version tout en suggérant (poussant) les modifications que l'on souhaiterait voir reprendre dans la version principale.

licences informe la liste et, si approuvée, la licence est ajoutée au site de l'OSI.

Une diffusion des bonnes pratiques au sein des structures. S'apercevant de l'importance de la dimension sociale de ces licences, les différentes structures qui forment le monde du Libre se sont par ailleurs mieux organisées, notamment dans la répartition de leurs tâches en privilégiant la complémentarité et la transparence. Au regard de son âge et de ses missions, la Free Software Foundation s'est ainsi rapidement appuyée sur un certain nombre de « structures sœurs » qui complétaient ses tâches : tels le Software Freedom Law Center (SFLC) créé en 2005 (réellement opérationnel à partir de 2007) qui assure, sous la direction d'Eben Moglen, l'assistance juridique aux organisations à but non lucratif et la diffusion de connaissances juridiques au travers d'une activité de publication, ou encore la Software Freedom Conservancy créée en 2006 pour aider à promouvoir, améliorer, développer et défendre des projets de logiciels libres et *open source* (tels des projets réputés comme Amarok, BusyBox, Git, Inkscape, JQuery, Samba ou Wine) sous la direction de Bradley Kuhn. Dans la même orientation, l'Apache Software Foundation héberge depuis 2002 au sein de l'Apache Incubator un nombre croissant de projets qui lui sont confiés pour bénéficier de ses processus de développement et de gestion (de projet et de licences) nécessaires à leur développement ouvert et collaboratif. De manière plus informelle, la Mozilla Foundation a lancé en 2011 l'initiative Web FWD (pour Web Forward) afin d'aider les projets orientés en faveur de technologies ouvertes du Web et la fondation Linux ouvre progressivement des sous-groupes industriels. Plus loin et conséquemment à l'introduction progressive des acteurs traditionnels, un nombre croissant d'organisations industrielles a vu le jour afin de mutualiser différentes missions avec des partenaires (clients ou fournisseurs) et concurrents au sein de fondations, associations, consortium ou alliances : tels l'association ObjectWeb, l'Eclipse Foundation, l'Open Handset Alliance, L'Alliance GENIVI, la Foundation LiMo.

La Fondation Eclipse. Initialement consortium formé par IBM lors de l'ouverture de la plate-forme Eclipse en *open source*, la structure est devenue Foundation Eclipse en février 2004 (avec, comme membres fondateurs des sociétés telles que Ericsson, HP, IBM, Intel, MontaVista Software, QNX, SAP et Serena Software). La fondation supervise le déve-

loppement de l'IDE *open source* Eclipse et aide à cultiver une communauté *open source* et un écosystème de produits et de services complémentaires autour d'Eclipse. Aujourd'hui, presque deux cents membres financent la fondation par leurs cotisations, ce qui lui confère toute son autonomie même si l'implication d'IBM et des autres membres stratégiques reste très forte. Chacun des membres stratégiques possède un représentant au conseil d'administration de la fondation Eclipse, ce qui leur permet d'avoir une influence directe sur la direction stratégique de la fondation. Les membres stratégiques possèdent aussi un siège dans les différents conseils de la fondation, fournissant ainsi leur avis sur les thèmes et les priorités de l'écosystème Eclipse. La fondation met à disposition de ses membres un certain nombre de documents juridiques (statuts, contrat d'adhésion, politique en matière de PI et de respect des dispositions antitrust) et techniques (processus de développement essentiellement) ainsi que la licence Eclipse déjà évoquée.

L'Alliance GENIVI. Il s'agit d'une alliance industrielle à but non lucratif et ayant pour objet de promouvoir l'adoption générale d'une plateforme *In-vehicle infotainment* (IVI, infodivertissement pour véhicule) de référence basée sur des solutions libres, avec un développement *open source*. Elle a été fondée le 2 mars 2009 par le Groupe BMW, Delphi, GM, Intel, Magneti-Marelli, PSA Peugeot Citroen, Visteon, et Wind River Systems et compte aujourd'hui plus d'une centaine de membres (des constructeurs automobiles, des fabricants de semi-conducteurs, des fournisseurs d'équipements et des fournisseurs de services et de logiciels). L'alliance a lancé en août 2011 son programme de certification qui permet aux membres de l'alliance de valider la conformité des plates-formes logicielles présentées à une version des spécifications d'architecture. Le 19 septembre 2012, GENIVI a ouvert officiellement son propre projet (relatif aux logiciels automobiles spécifiques développés par l'alliance) *open source* hébergé par la Fondation Linux.

ObjectWeb. ObjectWeb est une communauté créée fin 1999 par Bull, France Télécom R&D et l'INRIA, autour du développement mutualisé de *middleware* (intergiciel) libres et *open source* et qui a connu un certain nombre de phases dans son développement : transformation en un consortium international en 2002, fusion avec Orientware en 2007 (ce qui donna l'actuel OW2) ainsi qu'avec l'Open Solutions Alliance en 2009.

L'association Tizen. Anciennement LiMo Foundation, elle fut créée en janvier 2007 et est composée de plus de 50 membres et organisations. La gouvernance du projet est « ouverture et transparence », le leadership et la prise de décision sont entièrement partagés. Les membres fondateurs sont Motorola, NEC, NTT DoCoMo, Panasonic Mobile Communications, Samsung Electronics, et Vodafone. Abandonnant son propre projet Meego (provenant de la jonction de Maemo de Nokia et Moblin d'Intel), Intel a rejoint la structure en 2011. Devenu Tizen Association, le projet est maintenant hébergé par la Fondation Linux. Le fonctionnement de ce projet est relativement intéressant puisqu'il repose à la fois sur la Fondation Linux qui assure la gestion du projet et sur une association dédiée au sein de laquelle les industriels se réunissent pour mutualiser leurs besoins, identifier les modèles économiques et travailler sur les services annexes (marketing et éducation notamment). Le choix de la Flora licence sur certains composants de la plate-forme est néanmoins critiqué au regard des limitations qu'apporte cette licence (conditionnant le bénéfice de certaine clause, notamment la licence en matière de brevets, au respect de certaine procédure ou test spécifique à la plate-forme).

L'Open Handset Alliance. Pour favoriser le développement du projet Android (système d'exploitation pour mobile et *framework* pour développer de nouveaux logiciels) diffusé sous licence Apache v2, la société Google a formé l'Open Handset Alliance¹ qui réunit les sociétés désireuses de participer à ce courant d'*open innovation* dans le secteur du mobile (elle est composée de 79 membres dont des opérateurs mobiles, des constructeurs de téléphones, de microprocesseurs, des éditeurs de logiciel et des distributeurs).

Conclusion

Le monde d'aujourd'hui n'est plus celui des années 1980. Il est évident que le logiciel libre est infiniment plus mature aujourd'hui qu'il ne l'était à ses débuts et que l'évolution des différents projets de logiciels libres a entraîné le perfectionnement de leur gouvernance. Les licences

1. L'Open Handset Alliance (OHA) est une alliance commerciale de 79 sociétés réunies pour développer des standards ouverts pour les terminaux mobiles. Voir <http://www.openhandsetalliance.com>.

libres eurent la double tâche de formaliser en interne les règles de contribution et d'agir en externe pour prévenir les risques identifiés au travers l'évolution des lois ou des pratiques (brevets, DRM, etc.).

L'apogée des licences libres, une ère révolue ? La multiplication de licences a permis de pérenniser et de renforcer ce courant en favorisant son adoption par des acteurs hétérogènes. En contrepartie, elle a engendré un certain nombre de problèmes (dont l'introduction de la notion d'incompatibilité et, plus généralement, une complexification du système) qui gênent la diffusion du mouvement, mais qui semblent néanmoins en passe d'être résolus par la mise à jour progressive des différentes licences (et l'arrivée de nouveaux schémas de licences telle que l'initiative *GPL.Next* de Richard Fontana), et par l'interconnexion croissante entre les différents projets (la circulation des contributions et des personnes favorisant une harmonisation des licences). Aujourd'hui, les licences suivent de plus en plus un processus d'édiction communautaire et leur nombre diminue, de sorte qu'il semble possible d'espérer que les préoccupations futures concernent d'avantages l'encadrement des communautés que le seul contrat assurant la gestion des droits. Elles resteraient ainsi tout aussi essentielles, mais en devenant mieux comprises, mieux admises et plus sécurisées. Le risque afférent s'en trouvera diminué, le sujet deviendra moins central et les obstacles à l'adoption moins importants (ce qui est d'autant plus dommage que les juristes commencent seulement à s'y intéresser). Au contraire, du côté des pratiques, des outils et de la gestion des communautés, les prochaines décennies seront certainement riches et cruciales. Relativement récents (tous deux lancés en 2010), les projets *Harmony Agreements* (qui permet la création de contrats de cession de contributions génériques) et le projet *Software Package Data Exchange (SPDX)*, sont de nouvelles réponses susceptibles de favoriser la diffusion du logiciel libre.

Une vigilance toujours nécessaire : les risques emportés par l'industrialisation et la nécessité de définition plus large. Grâce au Libre (et aux multiples courants associés), le logiciel devient ainsi plus proche d'une connaissance qu'on partage que d'un produit que l'on vend. Cela favorise un rapport d'égal à égal entre les individus, les communautés, les entreprises et les institutions publiques. Le prochain défi auquel devra peut être faire face le Libre concerne d'une part les pratiques indus-

rielles, de plus en plus fines, qui se développent autour d'une industrialisation de projets majeurs diffusés sous licences *open source* et, d'autre part, des évolutions techniques qui tendent à rendre le logiciel *in fine* beaucoup moins visible pour l'utilisateur final. Concernant l'industrialisation du Libre/de l'*open source*, il convient en effet de constater que, si les licences sont généralement tout à fait conformes aux définitions FSD ou OSD, l'utilisateur se retrouve de moins en moins bénéficiaire des libertés dont il devrait disposer. Il conviendrait même de se demander si la structuration de l'écosystème par la réunion d'entreprises autour de projets majeurs peut aboutir à des situations d'ententes mais aussi à des abus de position dominante (les deux étant condamnables sur le plan du droit de la concurrence, mais il nous intéresse essentiellement de percevoir l'impact sur les utilisateurs et sur les logiciels eux-mêmes).

L'exemple le plus symptomatique concerne sans aucun doute l'écosystème construit autour du projet Android qui réunit près d'une centaine de sociétés au sein de l'Open Handset Alliance et confère une place prééminente à Google en tant que titulaire des marques, mais surtout en qualité d'éditeur du logiciel et de fournisseurs des services (dont l'accès au Market ou encore un support de niveau très élevé au bénéfice des fabricants). Ainsi, l'éditeur a pu faire renoncer (au motif d'une compatibilité dont il définit les termes) aux libertés assurées par les licences du logiciel à un certain nombre de grands constructeurs, tel Acer qui dû renoncer en 2012 à la distribution en Asie de terminaux fonctionnant grâce à Aliyun OS (*fork* chinois d'Android développé par AliCloud, une filiale du moteur de recherche Alibaba) ou encore Motorola et Samsung qui souhaitaient en 2010 intégrer le service de localisation de la société Skyhook à la place de celui offert par Google. Les libertés deviennent dès lors relatives, organisant un statut *open until it matters*¹ bien loin des objectifs des millions de contributeurs adoptant sciemment des licences comme la GNU GPL. De telles situations réduisent le bénéfice des libertés assurées par les licences et il serait important de conserver un regard critique sur ces comportements.

L'un des chantiers des prochaines années sera ainsi peut-être de repenser les libertés des utilisateurs à une autre échelle que le seul logiciel, premièrement, en étendant ces définitions (l'Open Definition vient par exemple

1. Que l'on pourrait traduire par « ouvert jusqu'à que cela compte (économiquement) ».

généraliser les principes pour les appliquer aux œuvres non logicielles et aux bases de données) et, deuxièmement, en les adaptant au changement de notre société (par exemple, pour ce qui concerne le *cloud*, les libertés des utilisateurs de tels services imposent une réflexion plus large¹). Les libertés d'aujourd'hui ne sont plus nécessairement tout à fait les mêmes que celles d'hier.

Références

- CLÉMENT-FONTAINE, Mélanie, « Les oeuvres libres », Thèse de doctorat, Université de Montpellier, 2006.
- CRAMER, Florian, « The Creative Common Misundestanding », dans : *FLOSS+ Art*, sous la dir. d'Aymeric MANSOUX et marloes DEVALK, OpenMute, 2008.
- JEAN, Benjamin, *Option libre. Du bon usage des licences libres*, Paris : Framasoft/Framabook, 2012.
- LESSIG, Lawrence, *Code and Other Laws of Cyberspace*, New York : Basic Books, 2000.
- « Enormously Important News From the Free Software Foundation », dans : *Blog de L. Lessig* (2008).
 - « How I Lost the Big One », dans : *Stories, Legalaffairs* (2004), URL : http://www.legalaffairs.org/issues/March-April-2004/story_lessig_marapr04.msp.
- STALLMAN, Richard M., « The GNU Operating System and the Free Software Movement », dans : *Open Sources : Voices from the Open Source Revolution*, sous la dir. de Chris DIBONA, Sam OCKMAN et Mark STONE, Sebastopol, CA : O'Reilly, 1999.
- « What is the Free Software Foundation ? », dans : *GNU's Bulletin 1* (1986), p. 8.

1. Sans être mature, le mouvement de l'Open Cloud n'est pas récent. Citons par exemple la fondation OpenWeb qui a publié plusieurs licences qui s'étendent tant au droit des brevets qu'au droit d'auteur ; l'Open Web Final Specification Agreement (OWFa 1.0) utilisé par le projet Open Compute, lancé en 2011 par Facebook ; etc. Il nous semble important d'intégrer à cette définition les problématiques d'accès aux données, de logiciels libres et d'interopérabilité.

WILLIAMS, Sam, Richard STALLMAN et Christophe MASUTTI, *Richard Stallman et la révolution du logiciel libre. Une biographie autorisée*, Paris : Eyrolles, 2010.

Libre choix ou choix du Libre ?

Primavera DE FILIPPI
Isabelle RAMADE

Cet article présente une analyse historique de l'évolution des licences libres qui visent à maximiser à la fois la liberté des auteurs et des utilisateurs (parfois au prix de restrictions comme la clause *copyleft*), vers des licences moins libres (dites ouvertes) qui visent à offrir une plus grande liberté de choix aux auteurs.

Internet et les technologies numériques ont bouleversé les modalités de production et de consommation de l'information, qui peut être désormais produite, reproduite, partagée et redistribuée extrêmement facilement, rapidement et à moindre coût.

Auparavant considérés comme des consommateurs passifs d'informations, les utilisateurs sont devenus des producteurs actifs de contenus. Le mouvement du logiciel libre est apparu lorsque les utilisateurs ont pris conscience du fait que le droit d'auteur représente une contrainte à la créativité et à la liberté des auteurs/utilisateurs souhaitant développer ou enrichir des logiciels préexistants. Avec l'arrivée de nouveaux dispositifs qui facilitent la production de contenus numériques et l'accès d'un public plus nombreux et divers à Internet, le mouvement du libre s'est étendu aux contenus littéraires, musicaux, audiovisuels, ou multimédias. Ainsi

émergent des licences libres applicables aux œuvres de l'esprit autres que les logiciels.

Au fil des années, le nombre de licences s'est multiplié. Certaines d'entre elles imposent des restrictions nouvelles aux utilisateurs, notamment en ce qui concerne la création d'œuvres dérivées et les modalités de leur réutilisation. Ces nouveautés répondaient aux besoins de certains auteurs extérieurs aux communautés initiales du Libre, réticents à mettre à disposition leurs œuvres sur Internet sans conserver un certain degré de contrôle sur leur réutilisation.

Ces licences permettent toutes la reproduction et le partage des œuvres de l'esprit. Cependant, une distinction émerge entre les « licences libres », qui n'imposent aucune restriction autre que l'attribution et – éventuellement – la clause *copyleft*, et les « licences ouvertes », qui permettent aux auteurs de choisir plus précisément les conditions d'exploitation de leur œuvre – tel le système de licences Creative Commons. Précisons que, dans le cas des contenus, la distinction libre/ouvert ne recouvre généralement pas la distinction libre/*open source* pour les licences de logiciels qui accordent en pratique les mêmes libertés.

Nous soutenons que ces deux types de licences, « libres » et « ouvertes », sont complémentaires et souvent utilisées pour des œuvres de natures différentes ou par des auteurs aux finalités différentes. Si les licences libres sont plus adaptées aux projets collaboratifs où la distinction auteur-utilisateur s'estompe, les licences ouvertes sont utilisées lorsque l'auteur maintient un lien fort avec l'œuvre ou lorsque la nature de l'œuvre justifie des restrictions sur sa modification ou sa réutilisation.

1. Genèse des licences libres

Avec l'avènement de l'informatique personnelle des années 1980, puis du Web dans les années 1990, les nouvelles opportunités offertes par Internet et les technologies numériques ont permis le développement de nouveaux modèles de production et d'échange non plus fondés sur la rareté, mais sur le partage et la coopération. D'une part, les technologies numériques facilitent la reproduction et favorisent le remix ou la création

d'œuvres dérivées à partir d'œuvres antérieures ; d'autre part, Internet encourage l'interaction et la collaboration d'individus dispersés à travers le monde, pour la production de logiciels ou de contenus, tout en permettant leur diffusion sur une échelle globale. Ces développements ont permis la création d'œuvres de l'esprit par un plus grand nombre de personnes, et contribuent progressivement à estomper la distinction entre auteurs et utilisateurs.

1.1 La révolution d'Internet : le numérique mis en réseau

Le monde numérique repose sur le codage binaire de l'information (composé de 0 et de 1). Celle-ci peut être produite directement sous forme numérique, ou numérisée à partir d'informations analogiques. Cela facilite la multiplication des contenus par la possibilité de créer des copies exactes (sans perte de qualité au cours du temps) et une réduction considérable des coûts de stockage. La neutralité de l'information numérique élimine la distinction entre différents types de contenus : qu'il s'agisse d'images, de textes ou de sons, ils peuvent être facilement reproduits, transmis et réutilisés sur différents supports et réseaux. Les technologies numériques permettent ainsi la création, en facilitant non seulement la reproduction, mais aussi l'extraction, l'assemblage, et le collage de contenus hétérogènes en contenus dits multimédia.

Ils peuvent être produits, grâce à des logiciels, par toute personne dotée de connaissances informatiques très élémentaires. Ils peuvent également être facilement diffusés (sur une échelle globale) par un transfert quasi-instantané des données numériques sur le réseau Internet. Étant donné que la reproduction des contenus en format numérique n'induit aucun coût additionnel, ils sont partagés au sein des communautés numériques encourageant la libre circulation des connaissances et des savoirs. L'apparition du Web dans les années 1990 a entraîné une expansion fulgurante d'Internet et l'utilisation massive des réseaux pair-à-pair (*peer-to-peer*). Il est désormais possible, pour tout individu connecté au réseau, d'accéder à tout moment à un nombre gigantesque de contenus qui peuvent se trouver n'importe où dans le monde : on passe ainsi d'une situation de rareté à une situation d'abondance culturelle.

Pourtant, Internet ne se limite pas à une bibliothèque d'informations. Le réseau facilite aussi la communication à distance et la création de communautés en ligne. Si les premières sont apparues dans les années 1970-1980 (*newsgroups*, *Internet relay chat*), elles se sont nettement développées dans les années 1990 grâce au développement d'outils participatifs sur le Web (comme les forums de discussions). De nouveaux contacts se créent grâce aux moyens de communication électronique (courrier électronique dès 1972, *chat* dans les années 1985 et suivantes, forums et groupes de discussion dans les années 1990, visioconférence dans les années 2000, et plus récemment, les réseaux sociaux). Ces contacts favorisent de nouveaux espaces de rencontre plus vastes et disjoints des espaces traditionnels. Dans les années 2000, les technologies qualifiées de Web 2.0 proposent des plates-formes et des outils innovants et simples (wikis¹, sites de partage de photos ou de vidéos, réseaux sociaux², etc.) permettant de coordonner les actions d'un très grand nombre de personnes réunies autour d'intérêts communs dans des domaines variés. Qu'ils soient amateurs ou experts, des individus avec des capacités et des objectifs différents peuvent ainsi collaborer sur un projet commun en contribuant de manière asynchrone, indépendamment du temps et du lieu où ils se trouvent.

À l'inverse des médias traditionnels, sur Internet l'utilisateur est devenu bien plus qu'un simple consommateur passif d'information. L'interactivité proposée par les technologies de l'information et de la communication ont transformé l'internaute en un agent qui s'implique directement dans le processus de consommation et de production des contenus en ligne (*user generated content*). Ainsi, à l'ère du numérique, la distinction entre auteur et utilisateur s'estompe progressivement. On voit apparaître une figure intermédiaire : l'*utilisauteur*, qui à la fois produit et consomme des contenus³.

La figure même de l'auteur évolue, elle est même remise en question, grâce aux nouvelles formes de collaboration offertes par les technologies

1. Le premier wiki a été créé en 1995 par Ward Cunningham.

2. On peut faire remonter les réseaux sociaux sur Internet à Usenet, mais les années 2000 marquent la généralisation de leur usage par le grand public grâce à des services de réseaux sociaux sur le web.

3. Pour plus de détails sur les transformations sociales et techniques à l'ère du web sémantique, voir Danièle BOURCIER et al., *Intelligent Multimedia : Managing Creative Works in a Digital World*, European Press Academic Publishing, 2010, p. 412.

numériques qui permettent la création d'œuvres collectives (ou collaboratives) à grande échelle¹, produites souvent de manière décentralisée par des communautés de pairs en réseau.

1.2 Les logiciels libres

Dans les années 1970-1980, il était pratique courante pour un groupe de développeurs/utilisateurs de collaborer au développement de logiciels informatiques et d'en partager les résultats avec la communauté encore restreinte des internautes.

La croissance de l'industrie informatique a favorisé la contestation de ces pratiques par certaines entreprises de plus en plus désireuses de réaliser une marge importante de profits sur la vente de leurs logiciels.

C'est en réponse à ces évolutions qu'il jugeait nuisibles que Richard Stallman a fondé en 1983 le projet GNU. Ce projet collaboratif avait pour objectif principal de promouvoir le développement de logiciels librement utilisables et modifiables par tous. Cela marque le début du mouvement du logiciel libre (*free software*), un terme englobant tous les logiciels qui peuvent être librement exécutés, étudiés, modifiés et redistribués, et dont le code source demeure toujours accessible. Afin de préserver ces libertés, Richard Stallman établit en 1985 la Free Software Foundation (FSF) et rédige, avec certains de ses membres, une licence qui est devenue aujourd'hui la plus populaire des licences libres : la licence publique générale GNU ou GNU General Public Licence (GPL) (1989)². Cette licence s'appuie sur le régime du copyright ou droit d'auteur suivant les pays. Elle en subvertit cependant l'esprit afin d'atteindre un objectif radicalement opposé : au lieu d'imposer des restrictions sur la reproduction et la réutilisation d'un logiciel, la GPL accorde par avance à tout utilisateur tous droits de reproduction et de réutilisation, avec, comme seule restriction imposée aux usagers, l'obligation de soumettre les œuvres dérivées

1. Une expression désigne de tels projets, les wikis en étant un archétype : « sites collaboratifs massivement multi-auteurs » (*Massive Multiauthor Collaboration Site*, MMC en anglais).

2. Pour une analyse plus détaillée de la licence, voir : Mélanie CLÉMENT-FONTAINE, « Droit des Créations Immatérielles », thèse de doct., Université de Montpellier I, 1999.

aux mêmes conditions que l'œuvre originale. C'est cette notion de *copyleft* (en contraste avec le *copyright*)¹ qui constitue aujourd'hui l'un des piliers fondamentaux du mouvement du Libre, sans toutefois en être une clause obligatoire. Depuis, de nombreuses licences libres ont été développées² – certaines *copyleft*, d'autres non – dans le but de satisfaire les besoins des différentes communautés numériques³.

Ainsi, bien que le logiciel ait été le plus tôt concerné par les licences libres, le mouvement du Libre a donné lieu à une idéologie⁴. Fondée sur les principes de liberté, de partage, et de bien commun, elle s'est rapidement étendue depuis le champ du logiciel vers tous les champs de la création.

1.3 Les œuvres libres (ou « contenus ») autres que les logiciels

Dans le cas des contenus autres que les logiciels, le mouvement du Libre a suivi à peu près la même évolution que celle du mouvement FLOSS⁵, mais à un rythme différent. Entre 1970 et 1990, lorsque la plupart des internautes provenaient du monde universitaire, militaire ou des entreprises d'informatique, de nombreux contenus étaient mis en ligne sur le réseau par l'intermédiaire de protocoles tels que FTP, Gopher, puis HTTP, sans se soucier d'une éventuelle protection par le droit d'auteur. Avec la popularisation du Web, les fournisseurs de contenus ont compris que

1. Le *copyleft* (versus *copyright*), parfois traduit en français par gauche d'auteur (versus droit d'auteur), s'appuie sur le *copyright* ou le droit d'auteur pour en abolir les restrictions tout en garantissant que toute œuvre dérivée sera elle aussi librement réutilisable. Pour plus de détails, voir : « Qu'est-ce que le *copyleft* » sur *GNU.org*. <http://www.gnu.org/copyleft>. Cependant, le terme est quelquefois utilisé (souvent incorrectement) pour faire référence à tout type d'activité favorisant le partage des œuvres de l'esprit, indépendamment du respect des quatre libertés fondamentales du logiciel libre (et notamment la liberté de créer des œuvres dérivées) et sans se soucier de la persistance de ces libertés (qui caractérisent le *copyleft*).

2. Une liste des licences de logiciels libres catégorisés selon plusieurs critères peut être consultée à l'adresse suivante : <http://www.gnu.org/licenses/license-list.fr.html>. Un schéma détaillé est aussi disponible sur : <http://www.gnu.org/philosophy/categories.html>.

3. Ces licences établissent des rapports différents entre les auteurs, les utilisateurs et l'œuvre. Pour une présentation détaillée des stratégies à adopter dans le choix d'une licence, selon les préférences des auteurs, voir Benjamin JEAN, *Option libre. Du bon usage des licences libres*, Paris : Framasoft/Framabook, 2012.

4. Plus d'informations sur <http://www.fsf.org/philosophy/free-sw.html>.

5. FLOSS est l'acronyme de *Free/Libre Open Source Software*.

d'énormes profits pouvaient être réalisés par la vente de ces contenus numériques, mais qu'elle était contrariée par le téléchargement massif *via* les réseaux pair-à-pair. Le corollaire fut une volonté croissante de lutter contre la contrefaçon (de musique, de films, etc.) par l'établissement d'un lobbying intense en faveur de lois de plus en plus répressives.

Les ayants droit ont ainsi commencé à faire valoir leurs droits sur Internet, tout en promouvant de nombreuses réformes législatives en matière de droit d'auteur ou de copyright qui, dans leur forme traditionnelle, n'étaient pas adaptées à l'ère numérique. Cela a donné lieu à d'importantes réformes, notamment les deux traités de 1996 de l'Organisation Mondiale de la Propriété Intellectuelle (OMPI) et leurs transpositions dans les droits nationaux¹. Elles sont cependant vues par certains auteurs comme un bouleversement de l'équilibre traditionnel du régime de leurs droits². Pourtant, la loi est devenue nettement plus favorable aux titulaires de droits, notamment en raison de la protection juridique accordée aux mesures techniques de protection³ et à la gamme réduite d'exceptions aux droits exclusifs accordés aux ayants droit⁴.

1. Voir le WIPO Copyright Treaty (WCT) sur le droit d'auteur, le WIPO Performances et Phonograms Treaty (WPPT) sur les interprétations et les phonogrammes (WPPT), tels qu'ils ont été transposés aux États-Unis par le Digital Millennium Copyright Act de 1999 (ci-après le DMCA), et dans la Communauté européenne par la directive 2001/29/CE sur l'harmonisation de certains aspects du droit d'auteur dans la société de l'information (ci-après la Directive sur la société de l'information), ainsi que la directive 96/9/CE concernant la protection juridique des bases de données.

2. Voir, entre autres : Lawrence LESSIG, *Free Culture*, New York : The Penguin Press, 2004 ; Pamela SAMUELSON, « The Copyright Grab », dans : *Wired* (1996) ; Jessica LITMAN, « Growing pains : adapting copyright for libraries », dans : sous la dir. de Laura A. GASSAWAY, *Education and Society* 63, 1997, chap. *New Copyright Paradigms*, p. 66–80 ; David LANGE et Jennifer L. ANDERSON, « Copyright Fair Use and Transformative Critical Appropriation », dans : *Conference on the Public Domain, Duke Law School*, 2001, URL : <http://law.duke.edu/pd/papers/langeand.pdf>.

3. Voir les articles 11 et 12 du WCT et les articles 18 et 19 du WPPT, transposés pour la Communauté européenne par les articles 6 et 7 de la Directive sur la société de l'information et aux États-Unis par la section 1201 du DMCA.

4. Voir l'article 10 du WCT et l'article 16 du WPPT ; transposés pour la Communauté européenne par la Directive sur la société de l'information qui introduit un régime différent d'exception pour les œuvres protégées technologiquement (article 5), avec une exception obligatoire pour les copies temporaires, ainsi qu'une liste de limitations facultatives que les États membres peuvent incorporer dans la législation nationale, sous réserve de la qualification de l'article 6(4), qui exclut de ce régime d'exception toute œuvre protégée, mise à la disposition du public à la demande, selon les dispositions contractuelles convenues entre les ayants-droits et les autres parties concernées ; et transposés aux États-Unis dans la sec-

L'une des conséquences de cette évolution juridique a été l'émergence d'un nouveau mouvement informel (le mouvement de l'*Open*). Sur le modèle du FLOSS, il se bat contre la réduction de la liberté dans la sphère privée des individus et de l'accès à la culture sur Internet par des initiatives visant à transposer les principes du logiciel libre dans le domaine des arts et de la culture¹. Ce mouvement est porté essentiellement par des communautés numériques – associations de logiciel libre, de défense des libertés sur Internet, communauté de Wikimédia ou d'autres projets collaboratifs sur le réseau, collectifs d'artistes, etc. – pour lesquelles les restrictions imposées par défaut par le régime du droit d'auteur représentent un obstacle à la liberté de création. Cela est dû, en partie, à la récente explosion de contenus produits non plus par des auteurs au sens traditionnel du terme (des individus qui produisent à plein temps, ou qui gagnent leur vie grâce à leurs créations), mais par des communautés d'*utilisateurs* qui produisent, individuellement ou collectivement, des contenus numériques de manière plus ou moins régulière afin de les publier sur le réseau. La plupart ne cherchent pas à maintenir l'exclusivité sur les contenus qu'ils produisent, mais cherchent plutôt à en maximiser la diffusion et, dans une certaine mesure, la réutilisation.

Ainsi, par analogie avec les licences de logiciel libre, de nouvelles licences ont été conçues, avec des dispositions plus adaptées aux contenus concernés². C'est le cas, par exemple, de la GNU Free Documentation License³, créée en 2000 par la Free Software Foundation, destinée aux manuels et à la documentation, et de la Licence Art Libre, introduite la même année, mais orientée plutôt vers les œuvres de nature artistique⁴, sans cependant exclure les autres types d'œuvres.

tion 1201(d) à (i) du DMCA, qui établit un régime d'exceptions beaucoup plus rigide que la doctrine traditionnelle de *fair use* prévue par le Copyright Act de 1976, 17 USC § 107.

1. Primavera DE FILIPPI, *Copyright Law in the Digital Environment : Private Ordering and the regulation of digital works*, LAP Lambert Academic Publishing, 2012.

2. Pour plus de détails en ce qui concerne l'application du *copyleft* aux œuvres de l'esprit autres que le logiciel, voir Antoine MOREAU, « Le copyleft appliqué à la création. Une reformulation des données culturelles ? », thèse de doct., Nice : Université de Nice Sophia Antipolis, 2001.

3. GNU Free Documentation License : <http://www.gnu.org/licenses/fdl.html>.

4. Licence Art Libre : <http://artlibre.org/licence/lal>. Cette licence est recommandée par la Free Software Foundation : « We don't take the position that artistic or entertainment works must be free, but if you want to make one free, we recommend the Free Art License ». Voir : <https://www.gnu.org/licenses/licenses.en.html>.

Bien que les motivations sous-jacentes soient semblables à celles du logiciel libre, dans le cas des contenus, l'accent est mis davantage sur les aspects de création et d'acquisition de notoriété, plutôt que sur l'aspect fonctionnel. L'utilisation de licences libres pour des contenus offre, en effet, deux types d'avantages fondamentaux.

Premièrement, en ce qui concerne la création des contenus, les licences libres facilitent la production d'œuvres complexes ou de grandes dimensions (tels que la documentation des logiciels informatiques, les manuels de programmation, etc.) dont la production nécessite souvent l'intervention d'experts opérant dans différents domaines. Ces licences favorisent aussi la création d'œuvres collaboratives ou participatives (l'archétype étant celui de Wikipédia, lancée en 2001), qui évoluent dans le temps grâce aux contributions d'un très grand nombre d'individus dont l'identité n'est pas déterminée à l'avance.

Deuxièmement, les licences libres permettent une plus grande diffusion des œuvres de l'esprit qui peuvent être librement reproduites et redistribuées sur une échelle globale. Cela est susceptible d'encourager non seulement le partage des connaissances et des savoirs (ce qui, dans le cas des manuels, aura aussi pour effet d'inciter à l'utilisation des logiciels les mieux documentés), mais aussi la diffusion virale (sur le modèle du bouche à oreille) des œuvres les plus appréciées par le public. Si la célébrité des auteurs est corrélée à la visibilité de leurs œuvres, sur Internet, la libre circulation des œuvres peut considérablement augmenter la notoriété des auteurs.

2. Du libre à l'ouvert ?

2.1 Diversification des licences

Alors que la section précédente a illustré les raisons qui ont motivé l'émergence d'un mouvement d'ouverture des contenus, celle-ci détaille l'apparition des licences qui ont accompagné ce mouvement jusqu'à aujourd'hui. En effet, face aux besoins émergents de différentes communautés de créateurs, qu'il s'agisse de chercheurs ou d'artistes (professionnels ou amateurs), de nouvelles licences se sont développées avec un degré de

restriction plus ou moins élevée selon les besoins spécifiques de différents auteurs ou *utilisateurs*.

Cette section présentera tout d'abord un aperçu (purement descriptif) de leur évolution et de leur diversification au cours des années, pour terminer par une synthèse de l'état de l'art en ce qui concerne les licences de contenus les plus utilisées de nos jours. Les problématiques concernant notamment la distinction entre licences « libres » et « ouvertes » et leurs finalités respectives, ne seront traitées que dans un second temps.

2.1.1 Aperçu historique

Les premières licences de contenus libres ont été conçues pour la mise à disposition des manuels et de la documentation informatique, justifiées par le fait que la valeur des logiciels libres ne pourrait pas être entièrement appréciée si les usagers n'ont pas les moyens d'apprendre à les utiliser. Ainsi, suite à la Free Document Dissemination License¹ (FDDL) développée en 1997 à l'INRIA par Bernard Lang², la GNU Free Documentation licence (GFDL) a été introduite par la FSF afin de favoriser la publication des manuels informatiques sous les mêmes conditions que les logiciels qu'ils décrivent. Avec des conditions semblables, la Common Documentation licence d'Apple, établie en 2001³, offre une alternative plus simple aux dispositions plutôt complexes de la GFDL.

En dehors du monde des logiciels, l'Open Content licence (1998) et l'Open Publication licence (1999) représentent les premières initiatives visant à adapter les principes du logiciel libre à d'autres types de contenus, notamment aux documents scientifiques. Adaptées au monde académique, elles visaient un partage plus facile des travaux des professeurs des chercheurs⁴. De même, la Design Science License (1999) est une licence

1. Le texte intégral de la FDDL est disponible sur <http://bat8.inria.fr/~lang/licence/v1/fddl.html>.

2. Bernard Lang est directeur de recherche à l'INRIA. Fervent défenseur de la philosophie *open source*, il est co-fondateur de l'Association francophone des utilisateurs de logiciels libres (AFUL).

3. La Common Documentation License est disponible sur <http://www.opensource.apple.com/cdl>.

4. L'Open Content License et l'Open Publication Licence ont été rédigées par David Wiley, professeur adjoint à l'Université de l'Utah. Plus de détails sur <http://www.opencontent.org>.

de type *copyleft* qui s'applique à tous types de contenus (à l'exception du code source des logiciels et de leur documentation) afin de faciliter l'avancement scientifique¹.

En France, les rencontres *Copyleft attitude* organisées par Antoine Moreau et al.² en janvier et mars 2000 à Paris dans deux lieux d'art contemporain, *Accès local* et *Public*, ont rendu possibles des discussions entre le monde du Libre, du droit et de l'art (artistes liés à la revue *Allotopie*). Ces discussions se sont poursuivies au fil des années par des rencontres régulières et sur la liste de diffusion *Copyleft Attitude*. Une conception commune du lien étroit entre le Libre et la création artistique en a résulté, concrétisée par une nouvelle licence, la Licence Art Libre³. Conçue pour promouvoir la création et la diffusion des œuvres sous des conditions plus favorables que celles prévues par défaut par le droit d'auteur, elle permet la libre reproduction et réutilisation des œuvres de l'esprit, tout en exigeant que toute création dérivée soit mise à disposition du public sous des termes identiques.

Plus orientées vers la musique, l'Ethymonics Free Music licence (2000)⁴, et l'Open Audio licence de l'Electronic Frontier Foundation (2001)⁵ sont deux licences inspirées de la logique *copyleft*, destinées à traiter spécifiquement les œuvres musicales. Le système de licences Open Music (2001) représente une autre tentative de reproduire la logique du logiciel libre dans le domaine de la musique, en prévoyant un ensemble de licences personnalisées (Vert, Jaune, Rouge et Arc-en-ciel) en fonction

1. Le texte intégral de la Design Science License est disponible sur <http://www.gnu.org/licenses/dsl.html>.

2. François Deck, Roberto Martinez, Antonio Galego et Emmanuelle Gall.

3. La Licence Art Libre a été créée par Mélanie Clément-Fontaine et David Geraud, juristes, Isabelle Vojdani et Antoine Moreau, artistes, avec les contributions de la liste de diffusion *copyleft attitude*.

4. Ethymonics était une petite société qui vendait de la musique libre. S'opposant à la présomption générale suivant laquelle le téléchargement libre de la musique nuirait à l'industrie musicale, cette licence a été conçue pour montrer comment la musique libre peut être utilisée comme un moyen de promotion des artistes. Bien que la société soit désormais fermée, le texte intégral de la Free Music License peut être visionné sur https://fedoraproject.org/wiki/Licensing/Ethymonics_Free_Music_License.

5. L'Open Audio Licence a été créée en 2001 par l'Electronic Frontier Foundation (EFF) pour encourager la libre utilisation et réutilisation des œuvres musicales. L'EFF encourage désormais les artistes à utiliser la Creative Commons Attribution-ShareAlike au lieu de l'Open Audio Licence.

de la nature des droits qu'elles concèdent¹. Elles représentent une étape importante dans l'évolution des licences de contenus, qui s'éloignent progressivement des principes du Libre par l'introduction de nouvelles restrictions portant atteinte aux quatre libertés fondamentales. Malgré leur succès initial, la plupart d'entre elles sont de nos jours devenues obsolètes, supplantées par le système Creative Commons.

Avec un nombre croissant de licences élaborées pour satisfaire les besoins toujours plus spécifiques des auteurs, les difficultés n'ont pas tardé à se manifester. Leur complexité et le manque de coordination entre les différentes initiatives a créé une situation de confusion concernant le choix des licences de la part des créateurs et leur compréhension de la part des utilisateurs. De plus, en contradiction avec leurs finalités, le manque d'interopérabilité entre elles risque de limiter la libre réutilisation des contenus. Certaines incompatibilités (notamment lorsque ces licences réclament le partage sous les mêmes conditions)², réduisent les opportunités d'assembler, d'intégrer ou de fusionner des contenus pour la création d'œuvres dérivées³. Enfin, dans la mesure où la plupart d'entre elles ont été rédigées selon la loi d'une juridiction spécifique, leur validité et applicabilité pourraient être remise en cause dans un contexte de portée internationale comme Internet⁴.

Depuis sa création en 2002, le système de licences Creative Commons s'est progressivement imposé, en proposant aux auteurs de combiner trois clauses à leur convenance (Non commercial, Pas de modifications, Partage à l'identique) en sus de la Paternité, afin de satisfaire les besoins de chacun. Nous consacrons la partie qui suit à détailler ce système en raison de l'importance qu'il a aujourd'hui acquise.

Creative Commons interdisant cependant l'introduction de restrictions supplémentaires, de nouvelles licences de contenus ont été développées au cours du temps : inspirées du modèle des licences Creative Commons,

1. Le système de licences Open Music permet aux ayants-droit de choisir parmi différentes options afin de sélectionner la licence qui satisfait au mieux leurs besoins. Voir <http://openmusic.linuxtag.org>.

2. Mélanie Dulong de ROSNAY, *Creative Commons Licenses Legal Pitfalls : Incompatibilities and Solutions*, Amsterdam : University of Amsterdam – Institute for Information Law (IViR), 2009.

3. Voir « la compatibilité au sein des licences libres », dans JEAN, *op. cit.*, p. 230.

4. Danièle BOURCIER et Mélanie Dulong de ROSNAY, *International Commons at the Digital Age*, Paris : Romillat, 2004, p. 182.

elles introduisent des clauses additionnelles interdisant certaines catégories d'usages. C'est le cas, par exemple, de la Creative Archive licence¹, utilisée au Royaume-Uni par la BBC et l'Open University.

2.1.2 Les licences Creative Commons

L'apparition des Creative Commons ne peut être comprise que par une analyse historique des évolutions juridiques, sociales et techniques qui ont accompagné l'adoption progressive du réseau², notamment l'extension des règles d'application du droit d'auteur sur Internet (extension de la durée de protection, réduction de la gamme d'exceptions, protection des mesures techniques de protection³) et la menace croissante des brevets logiciels.

C'est ainsi qu'en 2001, Lawrence Lessig crée l'organisation Creative Commons, avec des collègues universitaires provenant de différentes institutions et grâce au généreux soutien du Centre pour le domaine public de l'université de Duke⁴. Cette organisation à but non lucratif vise à faciliter la diffusion et le partage des œuvres de l'esprit⁵. Siégeant à Mountain View en Californie, ce projet réunissait à l'époque une petite équipe de personnes poussées par des motivations différentes et de natures souvent personnelles. Trois personnalités se détachent, qui ont eu une influence majeure sur l'évolution de l'organisation.

Lawrence Lessig, alors professeur à la *Stanford Law School*, était particulièrement intéressé par l'interaction entre le droit, le code, les normes sociales et le marché dans la régulation du réseau et le respect des droits

1. Créée en 2005, la Creative Archive License est une licence basée sur la licence BY-NC-SA de Creative Commons, interdisant tout usage promotionnel ou diffamatoire des contenus. Plus d'informations sur <http://www.bbc.co.uk/creativearchive>.

2. Pour une analyse détaillée du contexte socio-politique et juridique dans lequel les licences Creative Commons se sont développées, voir notamment Philippe AIGRAIN, « Les Licences Creative Commons : Origine, Développement, Limites et Opportunités », dans : *Sopinspace*, 2005, URL : http://vecam.org/IMG/pdf/etude_Sopinspace_Creative_CommonsV1.pdf.

3. En Anglais, *Digital Rights Management* (DRM).

4. Duke's University Center for the Public Domain, <http://web.law.duke.edu/cspd>.

5. Creative Commons est une organisation à but non lucratif qui a pour objet de sensibiliser le public aux problématiques actuelles du droit d'auteur tout en accompagnant les nouvelles pratiques de création à l'ère numérique. Plus d'informations sur <http://www.creativecommons.org>.

fondamentaux des internautes (et cela aussi bien dans une perspective juridique que philosophique). Ayant largement travaillé sur les biens communs informationnels, il tente, avec Creative Commons, de transposer le modèle des logiciels libres dans le domaine des œuvres de l'esprit de nature plus traditionnelles, telles que les œuvres littéraires, les arts plastiques et la musique.

James Boyle, actuellement directeur du Centre pour le domaine public de l'Université de Duke, a été l'un des premiers auteurs à dénoncer les dangers liés à la privatisation du domaine public et à développer un agenda positif visant à lui assurer une reconnaissance et une protection légale au niveau de la loi même¹. Sa participation au sein de l'organisation était donc motivée par une volonté de développer une approche juridique commune pour fédérer les initiatives visant à promouvoir et à valoriser les biens communs informationnels.

Enfin, Yochai Benkler, juriste et actuellement professeur à la Harvard Law School, se concentrait en particulier sur l'étude du réseau Internet, en analysant notamment l'interdépendance qui existe entre l'infrastructure physique de télécommunication, l'architecture logique (les logiciels du réseau) et les données qui circulent sur le réseau (les contenus). Sa contribution à l'organisation portait notamment sur l'analyse de l'auto-production par les pairs (*commons-based peer production*²) qui représente l'une des principales sources de biens communs informationnels.

Alors que l'organisation Creative Commons n'était initialement impliquée que dans des activités de sensibilisation et de communication, ainsi que dans la formation de partenariats stratégiques, l'équipe s'est rapidement dirigée vers une nouvelle direction. C'est ainsi que, à peine un an après sa création en 2002, Creative Commons élabore et publie un système de licences visant à réduire les obstacles juridiques à la libre circulation des œuvres de l'esprit, par le biais d'outils juridiques et techniques simples, efficaces et standardisés.

1. James BOYLE, « The Second Enclosure Movement and the Construction of the Public Domain », dans : *Law and Contemporary Problems* 66.1-2 (2003), p. 33-74, URL : <http://www.law.duke.edu/pd/papers/boyle.pdf>.

2. Yochai BENKLER et Helen NISSENBAUM, « Commons-based Peer Production and Virtue », dans : *Journal of Political Philosophy* 14.4 (2006), p. 394-419.

Née de la volonté de s'opposer aux restrictions croissantes du régime du droit d'auteur¹, et notamment le *Sony Bono Act* (surnommé le *Mickey Mouse Protection Act*) voté en 1998 par le Congrès et visant à prolonger de 20 ans la durée de protection du copyright aux États-Unis (juste avant que le personnage de Mickey ne tombe dans le domaine public), le système de licences Creative Commons permet aux auteurs d'accorder par avance certains droits au public afin qu'ils puissent être exercés sans demander d'autorisation. Ces licences s'adressent donc spécifiquement aux auteurs qui désirent partager leurs œuvres, afin d'en accroître la visibilité, et qui trouvent un intérêt à enrichir le patrimoine culturel de contenus librement accessibles. Alors que le régime du droit d'auteur classique accorde aux auteurs une exclusivité sur la totalité de leurs droits (« tous droits réservés »), ces licences les encouragent à n'en conserver qu'une partie (« certains droits réservés »). Le but est de réduire les répercussions négatives qu'un régime de droits exclusifs excessivement sévère a sur la libre circulation des savoirs, l'accès public à la culture et la créativité des auteurs.

Au fil des années, l'organisation Creative Commons s'est étendue dans le monde entier avec un réseau qui compte aujourd'hui plus de 100 affiliés dans 72 juridictions². Les licences ont, elles aussi, évolué pour mieux s'adapter au nouveau contexte dans lequel elles opèrent (en partant de la version 1.0 en 2002, à la version 2.0 en 2004, la version 3.0 en 2007, et d'autres versions futures). Elles résolvent, en partie, les difficultés rencontrées du fait de la profusion, de la complexité et des incompatibilités des licences antérieures³.

L'adoption des licences par le public a été relativement rapide : avec à peine 1 million d'œuvres en 2003, on est passé de 4.7 millions en 2004 à

1. Lawrence LESSIG, « CC in review : Lawrence Lessig on supporting the Commons », 2005. URL : <http://creativecommons.org/weblog/entry/5661>.

2. Depuis 2004, l'institution affiliée à Creative Commons en France est le Centre d'Études et de Recherches de Sciences Administratives et Politiques (CERSA), une unité mixte du CNRS associée avec l'Université Paris II Panthéon-Assas, qui a transposé les licences en droit français. Plus de détails sur <http://www.creativecommons.fr>.

3. Pour un aperçu historique de l'évolution des licences, voir <http://creativecommons.org/about/history>.

20 millions d'œuvres en 2005, pour arriver en 2012 à plus de 500 millions d'œuvres licenciées sous l'une des licences Creative Commons¹.

Bien que la plupart de ces licences possèdent des caractéristiques communes (la libre reproduction et distribution, *a minima* non commerciale, des œuvres dans leur forme originale)², elles diffèrent cependant selon les degrés de libertés qu'elle confèrent aux utilisateurs afin de satisfaire les sensibilités diverses des auteurs.

Les licences CC aujourd'hui répandues sont fondées sur quatre critères ou clauses de base : attribution (BY) ; pas d'utilisation commerciale (NC) ; pas de modification (ND) ; et partage sous les mêmes conditions (SA). Ces critères peuvent être combinés afin de produire jusqu'à six licences différentes³ : BY, BY-SA, BY-ND, BY-NC, BY-NC-SA, BY-NC-ND. De ces six licences, seulement deux (BY et BY-SA) sont des licences libres au sens du logiciel libre.

Chacune d'entre elles et leurs outils sont exprimés de trois manières différentes. Tout d'abord, en format textuel, par un contrat détaillé, destiné essentiellement aux juristes et un résumé explicatif compréhensible par tous. D'un point de vue technique, un fragment de code informatique (en format RDF/HTML) est généré automatiquement en fonction des critères choisis par l'auteur, de manière à pouvoir l'insérer directement dans les pages web où figurent les œuvres, et en faciliter la recherche ou l'identification par les moteurs de recherche. Enfin, les utilisateurs sont informés de tous les droits et devoirs afférant à une œuvre grâce à un système de badges (qui peuvent directement être incorporés dans une ressource numérique) illustrant clairement les conditions de réutilisation qui caractérisent chacune d'entre elles.

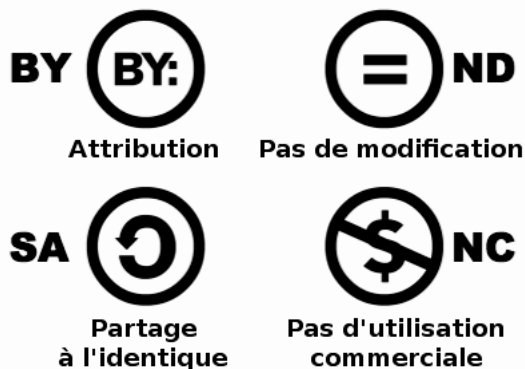
Ainsi, le système proposé par Creative Commons parvient à concilier le désir de diversité des auteurs (qui veulent maintenir un degré plus ou moins élevé de contrôle sur la réutilisation de leurs œuvres)⁴ avec le besoin de simplicité exprimé tant par les auteurs que par les utilisateurs.

1. Pour des informations plus détaillées sur les statistiques d'utilisation des licences Creative Commons, voir http://wiki.creativecommons.org/Metrics/License_statistics.

2. Voir http://wiki.creativecommons.org/Baseline_Rights.

3. Plus de détails sur les conditions des licences, sur <http://creativecommons.org>.

4. Shun-ling CHEN, « To Surpass or to Conform. What are Public Licenses For ? », dans : *University of Illinois Journal of Law, Technology & Policy* 1 (2009), p. 107-139.



Les quatre clauses et leurs logos

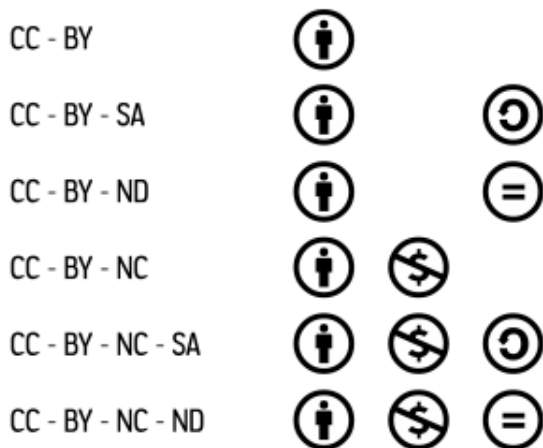
En outre, Creative Commons a conçu des licences spécifiques à certaines catégories de contenus ou à des usagers particuliers, telles que la Music Sharing licence et les licences Sampling, Sampling+ et Non-commercial Sampling+ pour promouvoir la diffusion et l'échantillonnage d'œuvres musicales ; la Public Domain Dedication et la Founders Copyright licence, pour renoncer à tous les droits d'auteur sur une œuvre après un certain nombre d'années ; et la Developing Nations licence, proposant des conditions moins restrictives pour les pays en voie de développement.

Par souci de simplification, Creative Commons a décidé de suspendre le développement de certaines licences trop spécialisées ou qui ne répondaient pas aux besoins d'un nombre suffisant d'auteurs¹, afin de se limiter à un système de six licences.

Inversement, pour répondre au besoin croissant d'identifier et de repérer les œuvres dans le domaine public, Creative Commons a introduit deux nouveaux outils : CC0², créé en 2009 pour que les auteurs ou les ayants

1. Certaines des licences proposées initialement par Creative Commons ont été retirées, car peu utilisées ou en conflit avec des valeurs considérées plus importantes. Une liste de ces licences, ainsi que les raisons pour lesquelles elles ont été abandonnées, est disponible sur <http://creativecommons.org/retiredlicenses>.

2. CC0 permet aux auteurs ou titulaires des droits d'auteur de renoncer à leurs droits afin de placer leurs œuvres dans le domaine public, dans la limite de la loi applicable (c.-à-d. où il est légal de placer de son vivant son œuvre dans le domaine public). Plus d'informations sur <http://creativecommons.org/about/cc0>.



Les six licences Creative Commons et leurs logos

droit puissent placer leurs œuvres dans le domaine public ; et la Public Domain Mark ¹, créée en 2010, pour faciliter la découverte et l'étiquetage des œuvres qui sont déjà libres de droits d'exploitation.

En ce qui concerne la validité des licences, Creative Commons travaille avec des experts en droit d'auteur dans le monde entier afin de les transposer aux spécificités locales de différents pays pour s'assurer de leur validité quelle que soit la juridiction.

En 2011, avec plus de 400 millions d'œuvres disponibles sur Internet sous l'une des licences Creative Commons ², ces licences sont désormais devenues une sorte de standard répandu de fait, en dehors de la volonté d'un organisme ou d'un consortium, et aujourd'hui reconnu en tant que tel par de nombreuses communautés numériques.

Or, malgré les efforts entrepris par Creative Commons pour faciliter le partage et la réutilisation des œuvres de l'esprit, il reste aujourd'hui des problèmes qui n'ont toujours pas été résolus (et qui sont, par nature,

1. La Public Domain Mark est un outil qui permet de marquer d'une façon simple et standardisée les œuvres qui ne sont plus protégées par le droit d'auteur, afin de les rendre facilement repérables et accessibles.

2. *The Power of Open*, disponible sur <http://thepowerofopen.org/>.

insolubles) : les incompatibilités entre les licences qui persistent au sein même du système de Creative Commons.

2.2 Licences libres et licences ouvertes

Alors que les licences libres ont été définies à plusieurs reprises par les communautés du Libre, il n'existe, de nos jours, aucune définition officielle, ni consensuelle, de ce qui constitue exactement une licence ouverte. Une analyse des licences de contenus présentées ci-dessus permettra d'illustrer plus clairement la différence entre les licences libres et ouvertes dans leurs diverses acceptions afin de mieux comprendre les raisons de leur diversification dans le temps ¹.

2.2.1 Les licences libres

Dans le domaine des logiciels, une première définition a été fournie par la FSF, identifiant les quatre libertés de base qui doivent être accordées par la licence pour qu'un logiciel puisse être considéré comme libre, et, plus tard, l'Open Source Definition de l'OSI décrira les conditions qu'une licence doit respecter pour être qualifiée de licence *open source* ². En dépit de leurs divergences quant à leurs principes (le libre au sens de la FSF est un mouvement philosophique et politique, alors que l'*open source* est une position pragmatique), ces deux définitions sont, en pratique, équivalentes : elles accordent les mêmes droits aux utilisateurs de tels logiciels.

1. Pour un aperçu des valeurs exprimées par différentes communautés numériques impliquées dans l'échange et le partage de contenus, voir Danièle BOURCIER et Primavera DE FILIPPI, « Les communautés numériques : objectifs, principes et différences », dans : *Cahiers Français* 372 (2012).

2. Plus de détails sur <http://opensource.org/docs/osd>.

Elles ont été reprises et adaptées au domaine des contenus par la Free Cultural Works Definition¹ et par l'Open Knowledge Definition² de l'Open Knowledge Foundation (OKF)³.

Inspirées des licences de logiciels libres, les licences de contenus libres confèrent aux utilisateurs un certain nombre de droits qui correspondent aux quatre libertés fondamentales définies par la FSF : le droit de reproduire, d'utiliser et de redistribuer une œuvre pour n'importe quelle finalité, ainsi que le droit de modifier une œuvre librement et de la redistribuer ainsi modifiée. Conformément aux principes du logiciel libre, ces licences n'admettent aucune forme de limitation à ces droits au-delà de l'attribution de l'auteur et, le cas échéant, les restrictions dues à la clause de *copyleft*. Cette dernière est aujourd'hui présente dans plusieurs licences de contenus libres, telles la GFDL, la Licence Art Libre, ainsi que la licence Creative Commons BY-SA. D'autres comme les licences de type BSD pour les logiciels ou la licence Creative Commons BY pour les contenus n'incorporent pas de clause *copyleft* afin d'en favoriser une réutilisation maximale.

2.2.2 Les licences ouvertes

Au cours du temps, les licences de contenus ont évolué avec des conditions de plus en plus sophistiquées, mais aussi de plus en plus restrictives.

1. L'objectif de la Free Cultural Works Definition est de déterminer les conditions selon lesquelles une œuvre et sa licence peuvent être considérées comme étant « libres ». Inspirée de la définition fournie par la FSF, la Free Cultural Works Definition identifie quatre libertés fondamentales qui doivent être accordées à tous pour qu'une licence puisse être considérée comme libre. Plus d'informations sont disponibles sur <http://freedomdefined.org/Definition>.

2. L'Open Knowledge Definition a été élaborée par l'Open Knowledge Foundation dans le but d'établir un ensemble de principes pour décrire ce qui constitue la « connaissance ouverte » (*open knowledge*). Ayant défini le terme « connaissance » de manière à inclure tout type d'information, de contenus ou de données, l'Open Knowledge Definition se concentre principalement sur l'explication du terme « ouvert » dans le contexte de la « connaissance ouverte ». La définition reprend notamment les 11 principes de l'Open Source Definition, les adapte le cas échéant aux spécificités des œuvres et des contenus autres que les logiciels. Plus de détails sur <http://www.opendefinition.org/>.

3. Alors que la première semble avoir été fortement inspirée par la définition du logiciel libre fournie par la FSF, la seconde reprend essentiellement les critères de l'Open Source Definition, tout en l'adaptant au domaine des arts et de la culture. Ainsi, bien que fondées sur des idéologies légèrement différentes, ces définitions décrivent exactement le même type de licences.

S'écarter ainsi des principes fondamentaux qui caractérisent le mouvement du Libre, ces licences permettent toutefois une plus grande liberté d'utilisation et de réutilisation des contenus par rapport au régime traditionnel du droit d'auteur. C'est le cas, par exemple, des licences Open Music (à l'exception de la licence verte), de certaines licences Creative Commons (BY-NC, BY-ND, BY-NC-ND et BY-NC-SA) et de la Creative Archive licence.

Ces licences se basent sur le même mécanisme de « détournement du droit d'auteur » que les licences de logiciel libre. Cependant, en raison des restrictions que certaines incluent, ces licences ne sont plus qualifiées de « licences libres », mais plutôt de « licences de libre diffusion » ou, plus généralement, de « licences ouvertes », un terme dont l'usage désormais très répandu en rend la définition nécessairement difficile à établir de manière univoque.

Ainsi, en France, le Conseil supérieur de la propriété littéraire et artistique (CSPLA) propose dans son rapport¹ de 2007 sur *La mise à disposition ouverte des œuvres de l'esprit* une définition minimaliste des licences ouvertes :

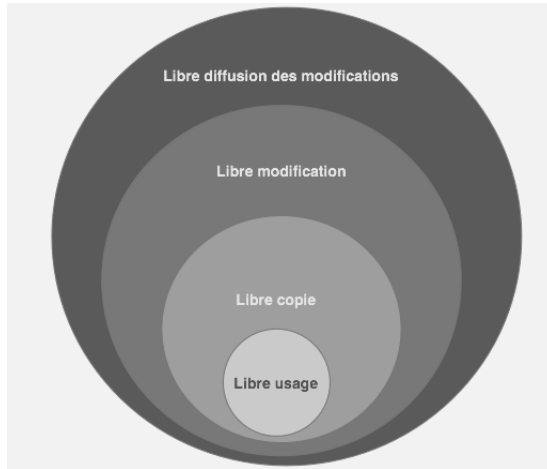
La Commission a retenu que le terme de « licence libre » devrait être réservé aux licences qui respectent strictement quatre libertés : libertés d'utiliser/d'usage, de copier, de modifier une œuvre et de diffuser les modifications. Par conséquent, une licence permettant l'exercice d'au moins une de ces libertés mais non nécessairement des quatre sera qualifiée, dans le présent rapport, de licence ouverte.

Néanmoins, cette définition n'est pas consensuelle et nous retiendrons par la suite une définition plus exigeante.

Il peut être utile de noter que la polysémie, suivant les contextes et les pratiques, du terme « ouvert » (*open*) explique l'absence de définition univoque de « licence ouverte ».

Par exemple, dans le contexte des initiatives d'*Open Access*, le terme est employé parfois *a minima*, comme désignant un accès ouvert gratuit (*Gratis Open Access*). En revanche, la déclaration de Budapest (BOAI)

1. <http://www.cspla.culture.gouv.fr/CONTENU/miseadiposouverterapp.pdf>.



D'après CSPLA, *La mise à disposition ouverte des œuvres de l'esprit*, 2007.

de 2002¹ concernant l'*Open Access* aux publications scientifiques le définit comme un accès gratuit assorti du droit de copie et redistribution² (*Libre Open Access*). Quant aux *Open Educational Resources* (OER)³, l'UNESCO en a proposé une définition⁴ qui exige une licence accordant les droits de redistribution, au moins non commerciale, et de modification ; certains entrepôts de ressources ou MOOC⁵ tels ceux du mouvement OpenCourseWare⁶ se conforment à cette définition.

Dans d'autres contextes, le terme « ouvert » est envisagé dans un sens plus étendu, synonyme de « libre ». Ainsi, l'Open Knowledge Foundation

1. Précisée par les déclarations de Bethesda et de Berlin en 2003.

2. On y trouve la phrase suivante : « By *open access* to this literature, we mean its free availability on the public internet, permitting any users to read, download, copy, distribute, print, search, or link to the full texts of these articles. »

3. *Open Educational Resources* (OER) a été traduit en français par Ressources Éducatives Libres (REL), Libre étant entendu dans un sens différent de celui de la FSF puisque la redistribution peut se limiter à une redistribution non commerciale.

4. « Les ressources éducatives libres sont des matériaux d'enseignement, d'apprentissage ou de recherche appartenant au domaine public ou publiés avec une licence de propriété intellectuelle permettant leur utilisation, adaptation et distribution à titre gratuit. » <http://www.unesco.org/new/fr/communication-and-information/access-to-knowledge/open-educational-resources>.

5. MOOC est le sigle de *Massive Open Online Courses*.

6. Lancé par le MIT, qui propose l'un des portails de ressources éducatives sous licence ouverte (CC-BY-NC-SA) les plus célèbres.

a donné de la connaissance ouverte (*open knowledge*) une définition correspondant à celle d'une connaissance sous licence libre et a établi une liste de critères¹ qui doivent être respectés par une licence pour que les contenus concernés puissent être qualifié d'ouverts : des critères qui rejoignent également ceux des licences libres.

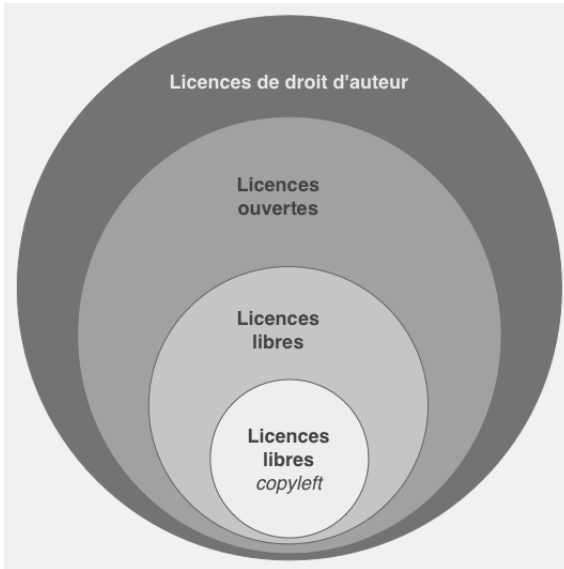
De même, le mouvement Open Data pour l'ouverture des données publiques, qui a émergé aux USA (data.gov) puis en Angleterre (data.gov.uk) a conduit à l'élaboration de licences pour des données ouvertes. En France, Etalab² a ainsi créé en 2011 une nouvelle licence nommée « Licence ouverte » qui promeut la réutilisation la plus large en autorisant la reproduction, la redistribution, l'adaptation et l'exploitation commerciale des données. On notera que ces libertés accordées font de cette licence une licence libre – l'adjectif ouvert est ici pris dans un sens maximal.

Ainsi, malgré la variété des droits accordés par les diverses licences dites *ouvertes*, il est néanmoins possible d'identifier les trois libertés de base qui se retrouvent dans la totalité de ces licences : toutes confèrent *a minima* le droit d'utiliser, de reproduire et de redistribuer des contenus à titre non-commercial, à condition que la paternité de l'œuvre soit citée (c'est cette acception que nous utiliserons par la suite). D'autres libertés peuvent s'ajouter à ces droits afin de créer une variété de licences conçues pour satisfaire les besoins de différents auteurs : le droit de créer des œuvres dérivées, le droit de réutiliser des contenus pour des finalités commerciales, etc. Selon le degré de liberté qu'elles confèrent aux utilisateurs, ces licences (plus ou moins ouvertes) se situent sur un *continuum* entre les licences libres et les licences exclusives de droit d'auteur et *a fortiori* le droit d'auteur dépourvu de licence.

Les licences ouvertes représentent donc une catégorie de licences bien plus large que la catégorie des licences libres (les dernières étant un sous-ensemble des premières). Le système de licences proposé par Creative Commons comporte aussi bien des licences « libres » que des licences

1. cf. <http://opendefinition.org>.

2. « Etalab est la mission, sous l'autorité du Premier ministre, chargée de l'ouverture des données publiques et du développement de la plate-forme publique *opendata* ». Définition de la Licence ouverte : <http://www.etalab.gouv.fr/pages/licence-ouverte-open-licence-5899923.html>.



D'après CSPLA, *La mise à disposition ouverte...*, 2007. Figure modifiée.

« ouvertes et non libres ». Cet éventail permet de satisfaire les besoins exprimés par une diversité d'auteurs. En effet, alors que la production d'un logiciel est généralement motivée par des finalités principalement fonctionnelles, et émane souvent de multiples contributeurs, la création d'une œuvre artistique ou littéraire est généralement poussée par des motivations autres et reflète plus profondément la personnalité des auteurs. Étant donné le lien qui les unit¹ à leur création, ces derniers sont alors plus susceptibles de vouloir conserver un certain degré de contrôle sur leurs œuvres afin d'en préserver l'intégrité. Les licences ouvertes établissent donc un compromis (entre la liberté de choix des auteurs et le degré de liberté conférée aux utilisateurs) qui permet de répondre aux désirs d'un plus grand nombre d'auteurs.

1. Le droit d'auteur classique des pays d'Europe continentale traduit une vision quasi-filiale du lien entre l'auteur et son œuvre.

3. Une question de liberté

Alors que les licences ouvertes maximisent la liberté des auteurs en leur offrant une plus grande liberté de choix, les licences libres visent à maximiser la liberté des *utilisateurs* – parfois au prix d’une limitation de liberté de chacun (*copyleft*) – en leur accordant le droit de réutiliser librement des contenus, et de créer de nouvelles œuvres à partir de contenus préexistants. *In fine*, l’intérêt des *utilisateurs* rejoint celui de l’œuvre elle-même et plus largement de la création : d’une part, on optimise l’accès à l’œuvre et, d’autre part, la redistribution et la génération d’œuvres dérivées assurent la pérennité de l’œuvre et lui permet de servir de terreau à la création ultérieure de nouvelles œuvres par d’autres artistes.

Ainsi, alors que les licences libres sont plus adaptées pour des projets collaboratifs où la distinction auteur-utilisateur s’estompe, et plus généralement dans les cas où l’auteur ne prime pas sur l’œuvre, les licences ouvertes sont utilisées lorsque l’auteur maintient un lien fort avec l’œuvre ou lorsque la nature même de l’œuvre justifie des restrictions à sa réutilisation ou à sa modification.

Nous proposons ici une réinterprétation des licences (libres et ouvertes) fondée sur une analyse de leurs finalités respectives. Si les deux ont pour but de garantir un certain degré de liberté, leurs finalités diffèrent en ce qui concerne le bénéficiaire de ces libertés : l’auteur (dans le cas des licences ouvertes) et les *utilisateurs* ainsi que l’œuvre elle-même (dans le cas des licences libres).

3.1 Liberté de l’auteur

Pour la plupart des contenus (artistiques, mais pas seulement), les œuvres représentent l’expression de la personnalité de leurs auteurs, ce qui légitime, pour le droit d’auteur classique, le droit de contrôler la réutilisation de leurs œuvres par des tiers. Cette idée se retrouve dans les fondements classiques du droit d’auteur¹, dérivés de la pensée naturaliste

1. Le code de la propriété intellectuelle confère des droits aux auteurs de « toutes les œuvres de l’esprit quels qu’en soient le genre, la forme d’expression, le mérite ou la destination » (CPI, art. L. 112-1). La seule condition nécessaire et suffisante pour bénéficier de la protection du droit d’auteur est le critère d’originalité, définie comme étant l’empreinte de la créativité et de la personnalité de l’auteur.

de John Locke et des théories de la personnalité développées par Kant et Hegel.

D'après John Locke, la propriété des biens (matériels et immatériels) n'est qu'une extension de la propriété de la personne. Une personne n'appartenant qu'à elle-même, elle incorpore dans son travail une partie de son être, et devient ainsi propriétaire du produit de son travail¹. Le droit d'auteur représente ainsi un droit naturel conféré aux auteurs en raison de l'effort intellectuel requis pour la production des œuvres de l'esprit.

D'autre part, les théories de la personnalité de Kant et Hegel reconnaissent aux auteurs le droit d'avoir leurs œuvres protégées comme une extension de leur personnalité. Pour Kant, le lien unissant l'auteur à son œuvre représente une partie intégrante de sa personnalité, qui doit donc être protégée en tant que telle². Pour Hegel, l'œuvre constitue une manifestation de la volonté de l'auteur qui subsiste même après que l'auteur s'est séparé du médium dans lequel l'œuvre a été incorporée³. Ces théories ont fortement influencé le régime du droit d'auteur, notamment en Europe continentale, et représentent l'une des bases fondamentales sur laquelle le régime des droits moraux est le plus souvent justifié⁴. Elles entraînent une vision de l'œuvre qui subsiste non pas comme une entité

1. « Aucun autre que lui-même ne possède un droit sur elle, le travail de son corps et l'ouvrage de ses mains lui appartiennent en propre. Il mêle son travail à tout ce qu'il fait sortir de l'état dans lequel la nature l'a laissée, et y joint quelque chose qui est sien. Par là, il en fait sa propriété. Cette chose étant extraite par lui de l'état commun où la nature l'avait mise, son travail lui ajoute quelque chose, qui exclut le droit commun des autres hommes ». John LOCKE, *Le second traité de gouvernement*, sous la dir. de J.-F. (trad.) SPITZ, PUF, 1994, p. 22-23.

2. D'après la théorie de la personnalité développée par Emmanuel Kant, une œuvre de l'esprit doit être protégée dans la mesure où elle constitue une extension de la personnalité de ses auteurs, qui ont donc le droit de décider comment des tiers peuvent disposer de cette œuvre. La théorie de Kant est à la base du régime du droit d'auteur en Allemagne, et en d'autres pays de l'Europe Continentale. Léon MICHOU, *Théorie de la personnalité morale et son application au droit français*, LGDJ, 1924.

3. Hegel reconnaît que les œuvres de l'esprit sont le résultat de l'effort créatif des auteurs qui représente l'expression de leur personnalité, et donc une extension de leur personne. Par conséquent, les auteurs doivent conserver le droit de contrôler l'utilisation qui est faite de leurs œuvres, et les droits les plus fondamentaux, tels que le droit d'en empêcher la mutilation ou une attribution erronée, doivent être inaliénables. Georg Wilhelm Friedrich HEGEL, *Principes de la philosophie du droit*, Paris : PUF, 2003.

4. Charles R. BEITZ, « The Moral Rights of Creators of Artistic and Literary Works », dans : *Journal of Political Philosophy* 13 (2005), p. 330-358.

en soi, mais comme une empreinte de la personnalité de l'auteur dont les intérêts sont ainsi perçus comme étant supérieurs aux intérêts de l'œuvre.

Les licences ouvertes, libres ou non libres, reposent nécessairement, pour être légales, sur le droit d'auteur : *in fine* c'est bien l'auteur qui choisit une licence, ou accepte de placer sa contribution à un projet collaboratif sous une licence choisie par d'autres. Cependant, alors que les licences libres manifestent un « choix du Libre » par l'auteur, qui accepte par ce choix une restriction de sa propre liberté concernant la détermination des conditions de réutilisation de l'œuvre, les licences ouvertes mais non libres reposent sur la vision classique de la primauté de l'auteur, puisqu'elles visent à maximiser le « libre choix » des auteurs en leur offrant la possibilité de spécifier les modalités de réutilisation de leurs œuvres, parfois au prix d'une réduction du degré de liberté accordé aux utilisateurs, et d'une limitation de la diffusion de l'œuvre.

Il arrive ainsi que certains auteurs qui se sentent fortement liés à leurs œuvres désirent en préserver l'intégrité (en interdisant la modification par des tiers) sans pour autant vouloir limiter la visibilité. Cette situation est courante dans tous les cas où l'auteur occupe une position centrale par rapport à l'œuvre. Ce peut être le cas pour des œuvres à forte composante artistique, dont la modification constituerait aux yeux de certains auteurs une atteinte à leur personnalité, ou de ressources éducatives pour lesquelles certains auteurs craignent que des modifications par des tiers introduisent des erreurs qui demeureraient associées à leur propre nom. L'interdiction de produire des œuvres dérivées se retrouve aussi dans le contexte d'œuvres historiques ou scientifiques dont le contenu n'a pas vocation à être modifié – tel est le cas des discours, ou des rapports scientifiques, pour lesquels des licences comportant une clause de non modification permet aux auteurs de maintenir l'intégrité des contenus tout en facilitant leur diffusion et réutilisation. Plusieurs licences ouvertes – telles les licences Creative Commons BY-ND et BY-NC-ND – permettent de satisfaire les besoins de ces auteurs.

De même, pour de nombreux artistes, il est (ou paraît) difficile de trouver un modèle économique autre que celui fondé sur la vente des droits d'exploitation commerciale. Ces auteurs s'opposent ainsi à la réutilisation commerciale de leurs œuvres par des tiers lorsqu'elle leur paraît entrer en conflit avec leur propre rémunération issue de la cession des droits

à un bénéficiaire de leur choix. C'est pour cette raison que les licences ouvertes comportant une clause non-commerciale (telles que les licences Creative Commons BY-NC, BY-NC-SA et BY-NC-ND) sont souvent utilisées par les auteurs qui produisent des œuvres pour lesquelles il y a une forte demande de marché et/ou qui ont intérêt à faciliter la diffusion de leurs œuvres afin de développer ou renforcer leur réputation pour obtenir ultérieurement des profits en transférant contre rétribution les droits d'exploitation commerciale à des tiers. Pour ces auteurs, maintenir l'exclusivité sur la réutilisation commerciale de leurs œuvres est le meilleur moyen pour en tirer profit : leur priorité étant le revenu et non pas la diffusion maximale de leurs œuvres.

Cela confirme l'idée que ces licences ne sont pas synonymes d'absence de rémunération. De nouveaux modèles économiques ont été développés (et mis en œuvre avec des succès divers) par de nombreux acteurs du numérique, tels que Magnatune¹ et Jamendo² qui fournissent de la musique librement téléchargeable sous une licence ouverte, tout en profitant de la vente des CDs ou des utilisations commerciales ; Pragmazic³, fournisseur de bornes de partage musical et littéraire ; Flatworld⁴ et Sesamath⁵, qui fournissent des livres sous licences libres ou ouvertes, proposés gracieusement en format numérique bien que la version imprimée soit soumise à rémunération ; et des artistes comme Radiohead et Nine-Inch-Nails qui ont licencié leurs œuvres sous des licences Creative Commons sans pour autant se priver des revenus dérivés de leur vente ou de leur exploitation commerciale. Enfin, depuis le 1^{er} janvier 2012, la SACEM a lancé un projet pilote permettant à ses membres de licencier leurs œuvres sous des licences Creative Commons comportant l'option NC afin de maximiser

1. Magnatune est un label indépendant qui propose le téléchargement de musique sous licence Creative Commons BY-NC-SA, et qui vend les CD à un prix variable, dont 50% est reversé à l'artiste. Plus de détails sur <http://www.magnatune.com>.

2. Jamendo propose de la musique sous licences ouvertes et en téléchargement gratuit, les auteurs étant rémunérés par les dons des utilisateurs, la vente de licences d'utilisations commerciales, ou les revenus publicitaires de Jamendo dont 50% leur sont reversés. Plus de détails sur <http://www.jamendo.com>.

3. Les bornes Automazic et Minimazic sont des points d'accès de consultation et de téléchargement d'œuvres culturelles sous licences ouvertes. Il s'agit du premier système de téléchargement légal et illimité dans les lieux publics. Plus de détails sur <http://pragmazic.net>.

4. Flatworld est le premier éditeur mondial de livres scolaires et de manuels sous licences libres et ouvertes. Plus de détails sur <http://www.flatworldknowledge.com>.

5. <http://sesamath.org>.

ser la diffusion de leurs œuvres dans une optique non-commerciale, tout en bénéficiant du système de répartition prévu par la SACEM pour toute utilisation commerciale ¹.

3.2 Liberté de l'œuvre

À l'ère du numérique en réseau, la figure de l'auteur est ébranlée de deux manières. D'une part, comme nous l'avons explicité dans la première partie, tous les internautes sont potentiellement des auteurs du fait de la facilité de création, ré-assemblage, modification et diffusion offerte par le numérique. D'autre part, dans les projets massivement collaboratifs, chaque auteur joue un rôle infime et donc ne peut plus se mettre en avant. Ainsi, avec les nouvelles opportunités offertes par le numérique, il est aujourd'hui de plus en plus difficile de distinguer l'auteur et l'utilisateur, puisque les deux se rencontrent souvent en cette figure hybride que l'on a définie comme *utiliseauteur*. En outre, même lorsque l'auteur conserve en partie son statut en imprimant sa personnalité sur une œuvre – dont il est le seul créateur par exemple – les promoteurs de l'art libre considèrent qu'il ne doit pas primer sur l'œuvre : le processus de création, envisagé comme un flux, ne doit rencontrer aucune entrave à son « libre cours » ².

Dans le domaine littéraire, artistes et sémiologues ont déjà mis en évidence le caractère ouvert intrinsèque des œuvres, du moins sur le plan de l'interprétation, le lecteur participant à la réalisation de l'œuvre en lui donnant du sens ³. Ce rapport entre l'œuvre et le public a été exploré en détail par l'école italienne, et notamment dans les années soixante par Umberto

1. Bernard Miyet, président du Directoire la Sacem, précise : « Cet accord démontre la volonté de la Sacem de s'adapter aux pratiques d'une partie de ses membres, notamment en ce qui concerne les usages numériques. C'est un plus pour les auteurs, compositeurs et éditeurs qui, s'ils le souhaitent, pourront ainsi assurer la promotion non commerciale de leurs œuvres dans un cadre juridique défini, tout en conservant la possibilité de bénéficier d'une rémunération juste et effective pour l'exploitation de leurs créations. »

2. MOREAU, *op. cit.*

3. Voir, notamment Philippe BOOTZ, « Approche sémiotique d'un certain art programmé : œuvre signe et méta-lecture », dans : *L'art a-t-il besoin du numérique ?*, Cerisy La Salle, 2004, URL : <http://archivesic.ccsd.cnrs.fr> ; François RASTIER, « La sémiotique des textes entre philologie et herméneutique : du document à l'œuvre », dans : *Between Philology and Hermeneutics, Proceedings of the 11th International Conference Series*, Nagoya University, 2010, p. 11–26.

Eco pour lequel « l'œuvre d'art est un message fondamentalement ambigu, une pluralité de signifiés qui coexistent en un seul signifiant »¹ et qui exige donc un effort créatif, une contribution de la part du public pour comprendre et interpréter le message communiqué par l'œuvre².

Les arts interprétés (musique, théâtre...) et *a fortiori* les arts en partie improvisés, font nécessairement intervenir une autre figure que celle de l'auteur, celle de l'interprète, dans la finalisation de l'œuvre. Bien avant le numérique, il a existé des œuvres volontairement non clôturées par leurs auteurs, autorisant leur complétion par d'autres pour les parties non écrites (ornementations, cadences de concertos en musique par exemple), improvisées en temps réel ou écrites par d'autres compositeurs. Au XX^e siècle sont explorées les possibilités d'intervention sur l'œuvre de l'interprète et du public³ avec les concepts d'art interactif, d'œuvre inachevée, ou encore d'œuvre ouverte⁴. Cependant, dans le monde analogique, les contingences matérielles imposaient généralement la clôture des œuvres (un livre doit être clôturé pour être imprimé) et donc une discontinuité dans la succession des créations.

Aujourd'hui, en rendant infiniment plus aisée la modification, la recomposition, le remix etc., le numérique induit une rupture qualitative dans le processus de création, qui devient continu dans l'espace (sur le réseau) et le temps. Il brouille les frontières des œuvres et redéfinit les rôles. D'une part, il permet une évolution continue des œuvres : *versioning* permanent, création d'œuvres dérivées de plus en plus éloignées, ou d'œuvres recomposées. D'autre part, grâce à l'interactivité fournie par les technologies numériques, l'utilisateur assume un rôle bien plus fondamental dans le processus créatif. La distinction entre auteur et utilisateur s'estompe dès lors que le public contribue directement à la réalisation de l'œuvre

1. Umberto ECO, *L'oeuvre ouverte*, Paris : Seuil, 1965.

2. « Toute œuvre d'art alors même qu'elle est une forme achevée et close dans sa perfection d'organisme exactement calibré, est ouverte au moins en ce qu'elle peut être interprétée de différentes façons, sans que son irréductible singularité soit altérée. Jouir d'une œuvre d'art revient à en donner une interprétation, une exécution, à la faire revivre dans une perspective originale. » *ibid.*, p. 17.

3. L'art interactif, qu'on peut faire remonter à Marcel Duchamps, s'est surtout développé dans les années 1960, avec des artistes et théoriciens comme Roy Scott (projet collaboratif faisant intervenir 300 spectateurs).

4. Voir, par exemple la 3e sonate de Pierre Boulez.

qu'il consomme¹ – dans les limites fixées par l'auteur. Mais les licences libres vont plus loin en autorisant le public non seulement à interagir avec l'œuvre mais aussi à la faire évoluer en apportant des modifications à l'œuvre (remix, création d'œuvres dérivées), au-delà de celles prévues par avance par l'auteur originel².

Par ailleurs, les projets massivement collaboratifs comme le développement de logiciels libres phares (tel Linux) ou l'écriture de Wikipédia, contribuent à l'effacement de l'auteur du fait même du très grand nombre de coauteurs.

C'est ainsi que la notion romantique de l'auteur en tant que génie individuel qui est à la base du droit d'auteur à la française fait désormais place à la notion de génie collectif, qui n'existe que grâce à la contribution d'un très grand nombre de personnes, dont l'identité n'est pas toujours explicite.

Ainsi, dans l'ère numérique, les œuvres uniques et finies créées par le génie d'un auteur cohabitent désormais avec des contenus produits de manière collaborative par un nombre indéfini d'individus ; des contenus qui évoluent dans le temps, qui se multiplient et qui se diversifient continuellement³. L'auteur se dissocie ainsi de son œuvre, qui assume des intérêts distincts orientés vers la pérennité et le développement de l'œuvre comme une entité en soi, autonome et séparée de son auteur originel.

Il convient de noter que la plupart des individus qui contribuent aux projets collaboratifs (de logiciel ou de contenu) se qualifient eux-mêmes souvent non pas comme des « auteurs » mais comme des « contributeurs » ; ou dans le cas de projets visant à la production de textes, des « rédacteurs » : ces termes sont révélateurs de la différence de statut entre les créateurs d'œuvres collaboratives et les créateurs d'œuvres empreintes de la personnalité de l'auteur.

Ainsi, le terme « rédacteur » et non « auteur » est souvent utilisé pour qualifier les contributeurs de Wikipédia puisque l'un de ses principes fon-

1. Marshall LEAFFER, « Protecting Author's Rights in a Digital Age », dans : *University of Toledo Law Review* 1.27 (1995), p. 1–12.

2. Primavera DE FILIPPI et Benjamin JEAN, « De la culture papier à la culture numérique : l'évolution du droit face aux médias », dans : *Implications Philosophiques* 18 (2012).

3. Jos de MUL, « Digital Material : Anchoring New Media in Daily Life and Technology », dans : sous la dir. de J. RAESSENS et al., Amsterdam : Amsterdam University Press, 2009, chap. The work of art in the age of digital recombination, p. 95–109.

dateurs est la neutralité de point de vue (c.-à-d. ne pas mettre en avant son propre point de vue mais exposer les différents points de vue sur une question). D'ailleurs, chacun peut avoir contribué à une fraction infime du texte, a pu voir sa contribution modifiée par d'autres et apparaît parfois dans la « liste des auteurs »¹ sous forme de pseudo ou d'adresse IP. Ce ne sont donc pas les « auteurs » qui sont mis en avant², mais le résultat : l'article qui résulte de la contribution individuelle d'un très grand nombre d'individus³.

Il faut toutefois nuancer la corrélation entre l'effacement de l'auteur et le choix du Libre. En effet, de nombreux artistes se réclamant de la culture libre font le choix de placer leurs œuvres sous une licence libre (LAL, CC BY ou CC BY-SA), sans cesser pour autant de se définir comme auteurs, singuliers, créant des œuvres portant l'empreinte de leur personnalité. D'ailleurs, ils ne renoncent pas à la mention de la paternité. Cependant, ils n'en déduisent pas une subordination de l'œuvre à leur personne qui pourrait justifier une licence non libre.

À la vision discrète de la création sous-tendant le droit d'auteur classique, les promoteurs de l'art libre opposent une vision de la création comme un processus continu : la création est envisagée comme un flux, ce qui légitime la levée des obstacles à la circulation des œuvres, à leur modification et à leur intégration dans des œuvres tierces. Selon Antoine Moreau⁴, l'auteur est traversé par sa création ; c'est un passeur héritant des œuvres antérieures et rendant possible les œuvres futures : l'œuvre prime. À ses yeux, seules les licences libres sont adaptées au processus de création. Elles traduisent en droit les nécessités, l'essence même de la création, et s'avèrent particulièrement pertinentes à l'ère du numérique en réseau dont elles prennent en compte les propriétés intrinsèques pour favoriser l'épanouissement de la création.

1. Cette liste figure en lien dans l'historique de l'article.

2. Bien que le logiciel Mediawiki (moteur de Wikipédia) génère automatiquement la « liste des auteurs » de chaque article, cette liste n'est accessible que *via* la page d'historique de l'article que seuls très peu d'utilisateurs visitent.

3. En ce sens, Wikipédia se distingue radicalement des encyclopédies classiques comme *Britannica* ou *Encyclopædia Universalis*, qui au contraire mettent en avant la signature des articles (nom, titres et travaux des auteurs) et se décrivent comme des encyclopédies d'auteurs : l'article expose un point de vue sur un domaine, relevant de telle école de pensée en SHS, en philosophie, etc.

4. MOREAU, *op. cit.*

Ainsi, les licences libres favorisent ces nouvelles formes de production en maximisant la liberté et l'autonomie de l'œuvre, imposant pour ce faire des contraintes sur les auteurs (notamment dans le cas des licences de type *copyleft*) afin de garantir que l'œuvre aura toujours l'opportunité d'évoluer et d'être redistribuée librement. Les licences libres défendent ainsi le principe de la primauté de l'œuvre sur la personne de l'auteur.

Déjà, en limitant la durée de validité des droits patrimoniaux et en admettant donc que l'œuvre entre un jour dans le domaine public, le droit d'auteur classique reconnaît implicitement que l'œuvre échappe légitimement à son auteur (en dépit du fait que les droits moraux perdurent de manière imprescriptible). Cette limitation est fondée sur deux éléments : d'une part, l'auteur est redevable à la société, et en particulier aux auteurs et œuvres antérieures, de la possibilité de création de son œuvre propre ; d'autre part, l'intérêt du public doit être pris en considération car il en va de l'intérêt de la société entière, notamment en ce qui concerne l'accès à l'éducation et à la culture. Il est donc légitime que l'intérêt du public, lorsqu'il est en conflit avec celui de l'auteur, prime après un certain nombre d'années.

En inversant le point de vue, on peut envisager le droit d'auteur comme une exception transitoire au domaine public, où l'œuvre règne première¹.

Nous interprétons ainsi les licences libres comme un moyen d'assurer la primauté de l'œuvre dans la période couverte par le droit d'auteur. Dans cette approche, l'œuvre est reconnue comme autonome, possédant une vie propre indépendante de l'auteur, dès sa création : elle échappe *de facto* instantanément à son auteur. C'est pourquoi nous qualifions cette propriété intrinsèque de « liberté de l'œuvre ». L'expression « œuvre libre » étant, elle, introduite par analogie à « logiciel libre ». Les licences libres prennent en compte cette vision en légalisant la libre diffusion des œuvres et la possibilité de les modifier et de les réutiliser (en tout ou en partie) pour la création d'œuvres nouvelles.

S'il est vrai que les licences ouvertes imposent des contraintes aux utilisateurs afin de maximiser la liberté de choix des auteurs, les licences libres peuvent imposer des contraintes aux auteurs afin de maximiser la

1. Cette conception du domaine public a été élaborée notamment dans le Manifeste pour le domaine public élaboré dans le contexte du réseau thématique européen sur le domaine public numérique COMMUNIA (disponible sur <http://publicdomainmanifesto.org/french>).

liberté des œuvres de s'épanouir et de se développer dans le temps. Les licences libres de type *copyleft* représentent en effet un compromis (entre la liberté des auteurs présents et futurs) afin de les rendre tous égaux en droits devant l'œuvre. L'objectif est d'accorder à tous (utilisateurs ou auteurs potentiels) les mêmes libertés dont on a soi-même bénéficié en tant qu'utilisateur ou auteur, en ne limitant ces libertés que dans le but de faire perdurer cette égalité.

Perspectives et conclusion

Les licences libres et ouvertes partagent un même but : permettre aux auteurs d'accorder *a priori* des droits étendus aux utilisateurs ou à d'autres auteurs. Elles se distinguent cependant selon leur centre d'intérêt. Pour les licences libres, l'œuvre est première : en cas de conflit entre les intérêts de l'auteur et les intérêts de l'œuvre, elles arbitrent en faveur des intérêts de l'œuvre afin d'en maximiser la diffusion et d'en optimiser les opportunités d'évolution dans le temps. Inversement, pour les licences ouvertes, l'auteur est premier : en cas de conflit entre les intérêts de l'œuvre et de l'auteur, elles arbitrent en faveur des intérêts de l'auteur afin de lui garantir une plus grande liberté de choix.

Même si ces licences, pour être légales, s'insèrent dans le cadre juridique actuel de la propriété intellectuelle¹ (en l'occurrence, le droit d'auteur et le copyright) appliquée au champ des œuvres de l'esprit (logiciels et contenus), elles la remettent indirectement en cause en accordant par avance des droits plus ou moins étendus aux *utilisateurs* ultérieurs.

Ainsi, par rapport aux licences ouvertes, qui accordent un rôle prééminent à la personne de l'auteur, les licences libres sont plus avancées dans la prise en compte des changements induits par le numérique en réseau, notamment en ce qui concerne l'effacement des distinctions auteur/utilisateur.

1. Les licences (libres ou ouvertes) viennent en complément du droit applicable en permettant aux auteurs de disposer plus facilement de leurs droits. Cette approche a parfois été critiquée comme comportant le risque de renforcer les principes fondamentaux du droit d'auteur, plutôt que de les attaquer. Séverine DUSOLLIER, « Propriété intellectuelle : entre l'art et l'argent », dans : sous la dir. d'Ysolde GENDREAU, Montréal : Thémis – Université de Montréal, 2006, chap. The Master's Tools v. The Master's House : Creative Commons vs. Copyright.

Néanmoins, les licences ouvertes et non libres peuvent être envisagées dans certains cas comme une étape vers la libération complète des œuvres, étape nécessaire pour certains auteurs souhaitant utiliser de façon assouplie le droit d'auteur sans pour autant passer d'ores et déjà au Libre. Dans d'autres cas, elles peuvent être envisagées comme une solution souhaitable et définitive, car appropriée à des applications spécifiques et aux préférences des auteurs.

Qu'elles soient libres ou ouvertes et non libres, les licences de contenus telles que les licences Creative Commons contribuent à la valorisation des biens communs informationnels en facilitant le partage et la diffusion de la culture dans la société. Inspirées du mouvement du logiciel libre, ces licences ont mené à l'émergence d'un mouvement distinct et indépendant – le mouvement de l'Open – qui a inspiré à son tour de nouveaux mouvements tels que l'Open Access, l'Open Data, et l'Open Government pour en citer quelques uns,¹ mais aussi l'Open Design et l'Open Hardware qui ont transposé ce mouvement d'ouverture dans le monde tangible, dont l'examen va cependant bien au-delà de la portée de cet article.

Références

- AIGRAIN, Philippe, « Les Licences Creative Commons : Origine, Développement, Limites et Opportunités », dans : *Sopinspace*, 2005, URL : http://vecam.org/IMG/pdf/etude_Sopinspace_Creative_CommonsV1.pdf.
- BEITZ, Charles R., « The Moral Rights of Creators of Artistic and Literary Works », dans : *Journal of Political Philosophy* 13 (2005), p. 330–358.
- BENKLER, Yochai et Helen NISSENBAUM, « Commons-based Peer Production and Virtue », dans : *Journal of Political Philosophy* 14.4 (2006), p. 394–419.
- BOOTZ, Philippe, « Approche sémiotique d'un certain art programmé : œuvre signe et méta-lecture », dans : *L'art a-t-il besoin du numérique ?*, Cerisy La Salle, 2004, URL : <http://archivesic.ccsd.cnrs.fr>.

1. Pour une analyse de ces nouveaux mouvements d'ouverture, voir Primavera DE FILIPPI et Danièle BOURCIER, « Génération Y et gestion publique : quels enjeux ? », dans : sous la dir. de Nicolas MATYJASIK et Philippe MAZUEL, Institut de la gestion publique et du développement économique, 2012, chap. Vers un nouveau modèle de partage entre l'administration et les communautés numériques.

- BOURCIER, Danièle et Primavera DE FILIPPI, « Les communautés numériques : objectifs, principes et différences », dans : *Cahiers Français* 372 (2012).
- BOURCIER, Danièle et Mélanie Dulong de ROSNAY, *International Commons at the Digital Age*, Paris : Romillat, 2004.
- BOURCIER, Danièle et al., *Intelligent Multimedia : Managing Creative Works in a Digital World*, European Press Academic Publishing, 2010.
- BOYLE, James, « The Second Enclosure Movement and the Construction of the Public Domain », dans : *Law and Contemporary Problems* 66.1-2 (2003), p. 33–74, URL : <http://www.law.duke.edu/pd/papers/boyle.pdf>.
- CHEN, Shun-ling, « To Surpass or to Conform. What are Public Licenses For ? », dans : *University of Illinois Journal of Law, Technology & Policy* 1 (2009), p. 107–139.
- CLÉMENT-FONTAINE, Mélanie, « Droit des Créations Immatérielles », thèse de doct., Université de Montpellier I, 1999.
- DE FILIPPI, Primavera, *Copyright Law in the Digital Environment : Private Ordering and the regulation of digital works*, LAP Lambert Academic Publishing, 2012.
- DE FILIPPI, Primavera et Danièle BOURCIER, « Génération Y et gestion publique : quels enjeux ? », dans : sous la dir. de Nicolas MATYJASIK et Philippe MAZUEL, Institut de la gestion publique et du développement économique, 2012, chap. Vers un nouveau modèle de partage entre l'administration et les communautés numériques.
- DE FILIPPI, Primavera et Benjamin JEAN, « De la culture papier à la culture numérique : l'évolution du droit face aux médias », dans : *Implications Philosophiques* 18 (2012).
- DUSOLLIER, Séverine, « Propriété intellectuelle : entre l'art et l'argent », dans : sous la dir. d'Ysolde GENDREAU, Montréal : Thémis – Université de Montréal, 2006, chap. The Master's Tools v. The Master's House : Creative Commons vs. Copyright.
- ECO, Umberto, *L'oeuvre ouverte*, Paris : Seuil, 1965.
- HEGEL, Georg Wilhelm Friedrich, *Principes de la philosophie du droit*, Paris : PUF, 2003.
- JEAN, Benjamin, *Option libre. Du bon usage des licences libres*, Paris : Framasoft/Framabook, 2012.

- LANGHE, David et Jennifer L. ANDERSON, « Copyright Fair Use and Transformative Critical Appropriation », dans : *Conference on the Public Domain, Duke Law School*, 2001, URL : <http://law.duke.edu/pd/papers/langeand.pdf>.
- LEAFFER, Marshall, « Protecting Author's Rights in a Digital Age », dans : *University of Toledo Law Review* 1.27 (1995), p. 1–12.
- LESSIG, Lawrence, *Free Culture*, New York : The Penguin Press, 2004.
- LITMAN, Jessica, « Growing pains : adapting copyright for libraries », dans : sous la dir. de Laura A. GASSAWAY, *Education et Society* 63, 1997, chap. New Copyright Paradigms, p. 66–80.
- LOCKE, John, *Le second traité de gouvernement*, sous la dir. de J.-F. (trad.) SPITZ, PUF, 1994.
- MICHOUD, Léon, *Théorie de la personnalité morale et son application au droit français*, LGDJ, 1924.
- MOREAU, Antoine, « Le copyleft appliqué à la création. Une reformulation des données culturelles ? », thèse de doct., Nice : Université de Nice Sophia Antipolis, 2001.
- MUL, Jos de, « Digital Material : Anchoring New Media in Daily Life and Technology », dans : sous la dir. de J. RAESSENS et al., Amsterdam : Amsterdam University Press, 2009, chap. The work of art in the age of digital recombination, p. 95–109.
- RASTIER, François, « La sémiotique des textes entre philologie et herméneutique : du document à l'œuvre », dans : *Between Philology and Hermeneutics, Proceedings of the 11th International Conference Series*, Nagoya University, 2010, p. 11–26.
- ROSNAY, Mélanie Dulong de, *Creative Commons Licenses Legal Pitfalls : Incompatibilities and Solutions*, Amsterdam : University of Amsterdam – Institute for Information Law (IViR), 2009.
- SAMUELSON, Pamela, « The Copyright Grab », dans : *Wired* (1996).

V

Études de cas

Mozilla : histoire d'une libération

Robert VISEUR

Après le démarrage du projet GNU en 1984 et l'apparition de Linux en 1991, le lancement du projet Mozilla a probablement été un des événements les plus marquants en matière de logiciels libres à la fin du XX^e siècle. Il s'agissait en effet d'une initiative pionnière en matière de libération d'un logiciel propriétaire, alors que l'implication des entreprises commerciales dans le développement de logiciels libres et *open source* s'est accélérée au cours des dix dernières années. Netscape en était l'initiateur. L'entreprise détenait à l'époque plus de la moitié du marché des navigateurs Web face à Microsoft. L'enjeu était donc de taille : permettre à Netscape de maintenir son rythme d'innovation et de pérenniser son activité face à un concurrent représentant un acteur central dans l'édition logicielle.

Cette décision de Netscape aurait été inspirée par l'essai d'Eric Raymond intitulé *La cathédrale et le bazar*, dans lequel l'auteur présentait son expérience en matière d'édition libre avec le logiciel Fetchmail et apportait un ensemble de bonnes pratiques, dont le « *Release early. Release often.* » (en français : « Publiez tôt. Publiez souvent. ») est resté célèbre aujourd'hui. Pour autant, la libération de Netscape n'a pas été un long fleuve tranquille puisque l'équipe de développement a finalement été

licenciée en 2003 dans sa totalité, après plusieurs années de déclin commercial. Ce n'est que plus tard que le logiciel Firefox, dérivé du Mozilla original, a connu le succès que l'on sait aujourd'hui.

Cette aventure a été parsemée de succès mais aussi d'échecs. La mise en place du projet a nécessité la recherche de solutions sur les plans organisationnel, juridique, technique mais aussi économique. L'étude du couple constitué par Netscape et Mozilla a dès lors toute sa place dans l'histoire du logiciel libre, et reste riche en expériences utiles pour le lancement de nouveaux projets. Nous proposons dès lors une étude chronologique de l'histoire de Netscape et du projet Mozilla qui lui est associé.

Nous analyserons un ensemble de facteurs de succès et d'échec, et essayerons de comprendre pourquoi l'organisation commerciale derrière le projet Mozilla n'a pu être pérennisée. Nous serons en particulier attentifs aux facteurs reconnus comme influençant, négativement ou positivement, le développement des logiciels libres et *open source*¹ : caractère plus ou moins restrictif de la licence, organisation et outillage du processus de développement, mécanismes de responsabilité au sein de la communauté, présence ou non d'un sponsor, internationalisation du projet, complexité et modularité du code source, etc. Notre étude s'appuie sur les connaissances de l'auteur mais aussi sur l'abondante littérature, notamment scientifique, dédiée à la société Netscape et au projet Mozilla.

Nous avons structuré cette étude en quatre parties. La première traite des débuts du Web commercial, de la naissance et de l'héritage du projet Mosaic, puis de l'essor de la société Netscape (1993-1997). La seconde analyse la libération du navigateur Netscape, la naissance du projet Mozilla et la victoire progressive du navigateur Microsoft Internet Explorer, qui clôt la première « guerre des navigateurs » (1998-2002). La troisième étudie la montée en puissance de Firefox, un navigateur allégé, sécurisé, modulaire et respectueux des standards (2003-2007). La quatrième partie aborde la naissance du Web mobile et la concurrence croissante des navigateurs basés sur Webkit, et en particulier la forte percée du navigateur Google Chrome / Chromium (2008 - aujourd'hui).

1. Voir Stefano COMINO, Fabio Maria MANENTI et Maria Laura PARISI, « From planning to mature : On the success of open source projects », dans : *Research Policy* 36.10 (2007), p. 1575-1586.

1. 1993 – 1997 : héritage de Mosaïc et percée de Netscape

La paternité du World Wide Web est attribuée au CERN¹. Tim Berners-Lee et Robert Cailliau y mettent en place les bases techniques pour un partage ouvert de l'information sur l'Internet². Ils s'appuient sur un système d'adressage (URI), un protocole réseau (HTTP) et un langage de formatage (HTML) permettant la navigation par liens hypertextes. Le CERN met le système en fonctionnement à Noël 1990. Il publie en outre une bibliothèque de code (*libwww*) ainsi qu'un navigateur en mode ligne. Le mérite d'avoir rendu le W3 accessible à un plus large public est souvent attribué au NCSA (www.ncsa.illinois.edu). En 1993, Marc Andreessen y présente en effet le navigateur Web en mode graphique Mosaïc. Ce dernier propose une expérience d'utilisation sensiblement améliorée et sera massivement diffusé sur le réseau.

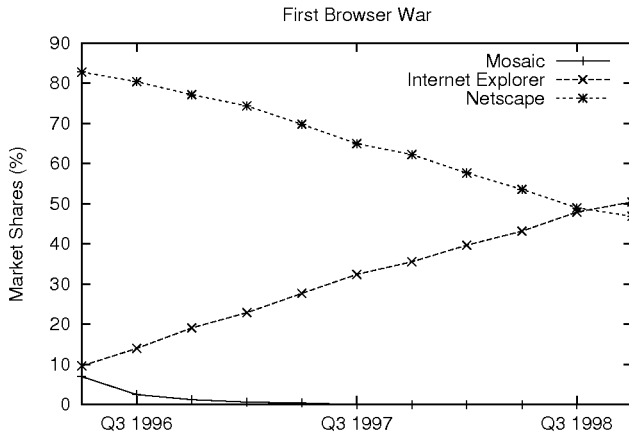
Une partie de l'équipe de développement de Mosaïc rejoindra la société Netscape, dont Marc Andreessen est co-fondateur. Netscape donnera naissance à une lignée de navigateurs populaires. Courant 1994, Netscape développe son offre de logiciels (clients et serveurs) pour le World Wide Web. Netscape tire alors ses revenus de la vente de licences.

Microsoft a pris du retard sur ce marché. Or, l'enjeu dépasse le seul marché des navigateurs Web³. Le risque existe en effet pour Microsoft que le développement de Netscape Navigator et du langage Java aboutisse à une plate-forme substituable au système d'exploitation Windows. Le premier navigateur de Microsoft, baptisé Internet Explorer, sortira à la mi-1995 et son premier serveur Web (Internet Information Server), début 1996. Microsoft ne partait pas de zéro puisqu'il pouvait se baser sur le code d'un navigateur licencié par la société Spyglass.

1. <http://info.cern.ch>.

2. Tim BERNERS-LEE et al., « The World-Wide Web », dans : *Communications of the ACM* 37.8 (1994), p. 76–82. Paul E. CERUZZI, « Aux origines américaines de l'Internet : projets militaires, intérêts commerciaux, désirs de communauté », dans : *Le Temps des Médias* 18.1 (2012), p. 15, URL : <http://www.cairn.info/resume.php?IDARTICLE=TDM0180015>.

3. Rached HALLOUL, « Le réseau stratégique et la concurrence illustrés par le cas M/N (Microsoft versus Netscape) », dans : *Innovations* 21 (2005), p. 197–216.



Évolution des parts de marché (Mosaic, Internet Explorer, Netscape) de 1996 à 1998 (données : EWS Web Server at UIUC, Wikipédia)

Netscape domine logiquement le marché jusqu'en 1997¹. Cependant, fin septembre 1997, Microsoft sort la version 4 de son navigateur, toujours diffusé comme *freeware* (graticiel) et d'une richesse fonctionnelle comparable au Navigator de Netscape.

Les forces de Netscape et de Microsoft étaient inégales. Microsoft comptait en effet plus de 18 000 employés (pour une gamme de produits nettement plus large), contre environ 1 000 chez Netscape. Le succès de ce dernier est attribué par Iansiti et MacCormack² à l'adoption par Netscape des techniques de développement flexibles. Dans un processus de développement flexible, les cycles de développement sont fortement raccourcis, de façon à permettre une correction rapide de la conception ou de l'implémentation. Les résultats sont très tôt soumis au verdict du client. Ainsi, les premières versions du navigateur Netscape sont réservées à un usage interne, de façon à ne pas frustrer les utilisateurs moins friands de nouveautés technologiques. Cependant, les versions suivantes sont pro-

1. Michael A. CUSUMANO et David B. YOFFIE, *Competing On Internet Time : Lessons From Netscape And Its Battle With Microsoft*, New York : Touchstone, 2000.

2. Marco IANSITI et Alan MACCORMACK, « Developing Products on Internet Time », dans : *Harvard Business Review* 75.5 (1997), p. 108–117, URL : <http://hbr.org/1997/09/developing-products-on-internet-time/sb1>.

posées en téléchargement aux clients. Le *feedback* des clients constitue une importante source d'informations pour l'équipe de développement. En particulier, parmi les clients testeurs, se trouvent les développeurs d'autres sociétés amenant le logiciel à ses limites et trouvant ainsi des bogues compliqués. Ce développement flexible a aidé Netscape à tenir tête à Microsoft jusqu'en 1997.

Microsoft a réussi en six mois environ à proposer un produit comparable à celui de Netscape. Microsoft a également eu recours à des techniques de développement itératif. Cependant, ces dernières étaient davantage tournées vers son Intranet, profitant de l'important nombre d'employés. Cette pratique consistant à utiliser ses propres produits pour en démontrer la qualité et les capacités est qualifiée de *dogfooding*, un néologisme dérivé de l'expression *eating your dog food*¹. Cette politique démontre une confiance certaine dans les produits de l'entreprise mais permet surtout la remontée des bogues vis-à-vis de scénarios d'utilisation réels. Surtout, Microsoft mise sur une stratégie d'intégration verticale. Internet Explorer est étroitement intégré au système d'exploitation Microsoft Windows et est fourni gratuitement avec ce dernier². Les utilisateurs sous Microsoft Windows, qui domine les systèmes d'exploitation pour ordinateur personnel, se montrent assez indifférents vis-à-vis de l'outil, et tendent à utiliser le logiciel installé par défaut.

2. 1998 – 2002 : libération de Netscape et victoire de Microsoft

Suite à la sortie d'Internet Explorer, Netscape réagit tardivement. Les parts de marché de Netscape diminuent rapidement depuis 1996. En janvier 1998, il annonce à son tour la gratuité de ses logiciels de navigation. Netscape annonce également la libération prochaine du code source de la version 5 de Communicator. Le code source est rendu disponible en

1. Warren HARRISON, « Eating your Own Dog Food », dans : *IEEE Software* 23.3 (2006), p. 5–7.

2. Tongsen WANG, Lei WU et Zhangxi LIN, « The revival of Mozilla in the browser war against Internet Explorer », dans : *Proceedings of the 7th international conference on Electronic commerce*, New York : ACM, 2005, 159–166.

mai 1998. Cette nouvelle branche *open source* coexiste avec l'ancienne branche propriétaire, dérivée de la version 4 du Communicator.

La version 5 de Netscape n'a pas connu d'avenir commercial. La version 6.0 est sortie tardivement à la fin 2000. Son instabilité, son incompatibilité avec de nombreux sites et, surtout, sa lourdeur ont déçu fans et utilisateurs. Il en résulte une perte de marché sensible. Le nombre de copies téléchargées de Netscape augmente régulièrement. Cependant, en terme de parts de marché, après avoir culminé à 90% environ à la mi-1996, les parts de marché de Netscape diminuent progressivement au profit de Microsoft, au rythme d'environ 1% par mois¹. De 18% début 2001, les parts de marché de Netscape sont passées à 12% mi-2001. La version 7.1, sortie en juin 2003 et basée sur Mozilla 1.4, a corrigé les défauts les plus gênants. Néanmoins, sur le plan commercial, le mal est fait. Cette victoire de Microsoft clôt le premier round de la bataille des navigateurs (*browser war*).

Jamie Zawinski², un des vétérans de Netscape et un des initiateurs du projet Mozilla.Org, a donné plusieurs raisons à cet échec initial de Mozilla.

1. Le projet était d'envergure et un délai important était nécessaire avant de pouvoir contribuer. En conséquence, une petite modification pouvait nécessiter plusieurs heures de travail.
2. Les contributeurs attendent une forme de rétribution, par exemple en ayant la capacité de modifier la version du navigateur couramment utilisée et d'ainsi voir les effets de sa propre modification. Or, les sources libérées ne sont pas celles de la version du navigateur commercialement distribuée (branche 4.x). Netscape a libéré un code auquel manquaient de nombreuses fonctionnalités et rempli de bogues.
3. Le code libéré était trop compliqué et difficile à modifier, si bien que peu de gens ont contribué. Une réécriture complète du navigateur a été nécessaire, ce qui a ramené le projet six à dix mois en arrière. Cette réécriture a donné notamment naissance au nouveau

1. CUSUMANO et YOUFFIE, *op. cit.* Serge DESCOMBES, « Saga Netscape/Microsoft : histoire d'un renversement », dans : *Journal du Net* (2002).

2. Jamie ZAWINSKI, *Resignation and postmortem*, www.jwz.org, 1999.

moteur de rendu : Gecko/Raptor. Ce besoin de réécriture a aussi été justifié par l'objectif de proposer un moteur de rendu respectant les standards ainsi que par une érosion du design, conséquence des pratiques de développement itératif¹.

4. Le code libéré ne contenait pas le lecteur de courriels, qui a aussi dû être réécrit à la suite de la sortie de Gecko/Raptor.
5. Les capacités de développement ont été partagées entre deux versions : la branche 4.x (elle a connu plusieurs évolutions, de la 4.5 à la 4.78) et la branche Mozilla. Le travail de codage effectué sur la branche propriétaire n'a pas bénéficié à la branche *open source* et en a ralenti le développement par dispersion des ressources.

Jamie Zawinski souligne néanmoins quatre apports positifs du projet.

1. Le projet Mozilla a apporté la démonstration qu'il était possible de gérer le développement distribué d'un logiciel de grande taille.
2. Le projet Mozilla n'a pas permis de récolter autant de contributions sous forme de code source que souhaité. Par contre, il a permis l'apport d'un feedback extérieur influençant la direction prise par les développements.
3. La libération de Netscape et la mise en place du projet Mozilla se sont accompagnées de la mise à disposition de précieux outils de développement. Ces outils sont utilisés pour le développement proprement dit du logiciel (XUL, Gecko, etc.) ou pour l'organisation du travail de collaboration sur le code source (Bugzilla, Tinderbox, etc.)².
4. Le projet Mozilla a contribué à la promotion et à la légitimation du modèle de développement *open source*.

En termes d'apport, nous pouvons aussi citer la création d'une nouvelle licence logicielle : la Mozilla Public License (MPL). Créée pour régler la cohabitation entre la branche communautaire et la branche commerciale

1. Christian Robotom REIS et Renata Pontin de MATTOS FORTES, *An Overview of the Software Engineering Process and Tools in the Mozilla Project*, Working paper, 2002. Jilles van GURP et Jan BOSCH, « Design erosion : problems and causes », dans : *Journal of System and Software* 61.2 (2002), 105–119.

2. REIS et MATTOS FORTES, *op. cit.*

du projet (cette dernière incluait plus de 75 modules tiers), la licence MPL est perçue comme originale par les juristes¹. Il s'agit d'une licence copyleft faible, qui se différencie notamment par la distinction faite entre les *œuvres modifiées* et les *œuvres plus larges*. Si les premières doivent rester sous licence MPL, les secondes peuvent par contre recourir à une licence tierce pour les parties originales. Sun Microsystems s'est inspiré de cette licence pour sa Common Development and Distribution licence (CDDL).

Le choix d'une licence logicielle pour couvrir le nouveau projet s'est avéré être un projet dans le projet². La licence BSD, très permissive, ne protégeait pas assez les apports des développeurs face à un usage sans contrepartie par un tiers. La licence GPL, plus restrictive (en ce sens qu'elle impose une conservation et une propagation de la licence), était jugée intenable pour les développeurs commerciaux. Une nouvelle licence, baptisée Netscape Public licence (NPL), a été soumise aux développeurs en mars 1998. Très mal accueillie, elle a été rapidement jugée inacceptable par la communauté compte tenu des droits spéciaux que Netscape s'y accordait. La NPL a ensuite débouché sur une seconde licence, baptisée Mozilla Public License (MPL), identique à la NPL mais faisant l'impasse sur la clause donnant la possibilité à Netscape de transférer du code sous licence NPL dans un produit non couvert par cette licence. Le code source du navigateur a par la suite évolué d'une couverture par la NPL originale vers une couverture par la MPL³. Les licences LGPL et GPL sont plus tard apparues dans le développement, de manière à régler les problèmes d'incompatibilités avec des projets tiers⁴.

Les difficultés liées à la libération de Netscape n'expliquent cependant pas tous les problèmes de l'entreprise Netscape. La stratégie adoptée par Microsoft pour contrer Netscape prendra aussi des aspects critiquables. Suite à l'échec d'un accord de partage de marché, l'entreprise de Red-

1. Yorick COOL et al., *Les logiciels libres face au droit*, Bruxelles : Cahier du CRID, Bruylant, 2005.

2. Jim HAMERLY et Tom PAQUIN, « Freeing the Source : The Story of Mozilla », dans : *Open Sources : Voices from the Open Source Revolution*, sous la dir. de Chris DIBONA, Sam OCKMAN et Marc STONE, Sebastopol : O'Reilly, 1999, p. 197-206, URL : <http://oreilly.com/openbook/opensources/book>.

3. Massimiliano DI PENTA et Daniel M. GERMAN, « An Exploratory Study of the Evolution of Software Licensing », dans : *ICSE'10 Mai* (2010).

4. Paul B. de LAAT, « Copyright or copyleft ? An analysis of property regimes for software development », dans : *Research Policy* 34 (2005), p. 1511-1532.

mond utilise « sa position dominante au sein de son réseau pour empêcher Netscape de vendre avec profit son navigateur »¹. Microsoft impose Internet Explorer à l'aide d'accords, par exemple avec les constructeurs informatiques, et s'appuie sur une politique de mise en incompatibilité systématique (guerre des formats)². Netscape se joint en 1998 aux actions en justice menées contre Microsoft pour pratiques anti-concurrentielles³. L'abus de position dominante est reconnu en 2000 (et confirmé en 2002) par la justice américaine. Pour Netscape, cependant, le mal est fait . .

En mars 1998, AOL Time Warner signe par ailleurs un partenariat stratégique avec Microsoft dans le but d'obtenir un navigateur personnalisé et la mise en évidence des services en ligne AOL. L'accord est symbolique. America OnLine (AOL) est en effet un acteur historique en matière d'accès Internet grand public et détient toujours une position centrale. Les conséquences sont importantes. Fin 1998, la part de marché d'Internet Explorer est de 43,8% mais tombe à 27,5% si on exclut AOL. Cet accord apporte à Microsoft un gain rapide de parts de marché. Selon Sebenius⁴, l'arrogance du management de Netscape, lui-même engagé dans ces négociations, estimant son produit incontournable du fait des faiblesses techniques de son concurrent, et le sens du compromis démontré par Microsoft pour atteindre des objectifs à long terme, contribuent à expliquer ce choix. AOL rachète finalement Netscape en mars 1999, autant pour son logiciel de navigation que pour bénéficier de l'audience du portail Netscape Netcenter. La part de marché du navigateur de Netscape tourne alors autour des 25%.

3. 2003 – 2007 : fin de Netscape et succès de Firefox

En juillet 2003, AOL Time Warner, propriétaire de Netscape depuis 1999, décide de licencier les derniers développeurs de Netscape. Le déve-

1. HALLOUL, *op. cit.*, p. 203.

2. Isabelle LIOTARD, « Les nouvelles facettes de la propriété intellectuelle : stratégies, attaques et menaces », dans : *Management et Sciences Sociales* 4 (2007), p. 1–14.

3. DESCOMBES, *op. cit.* HALLOUL, *op. cit.* Sandeep KRISHNAMURTHY, « Mozilla vs. Godzilla – The Launch of the Mozilla Firefox Browser », dans : *Journal of Interactive Marketing* 23.3 (2009), p. 259–271. WANG, WU et LIN, *op. cit.*

4. James K. SEBENIUS, « Negotiating Lessons from the Browser Wars », dans : *MIT Sloan Management Review* (2002).

lancement du navigateur Netscape est donc arrêté (même si plusieurs logiciels estampillés Netscape sont sporadiquement sortis après 2003). Mozilla, soutenu par la fondation du même nom (dont la pérennité n'est pas, à sa création, évidente), continue par contre d'être développé.

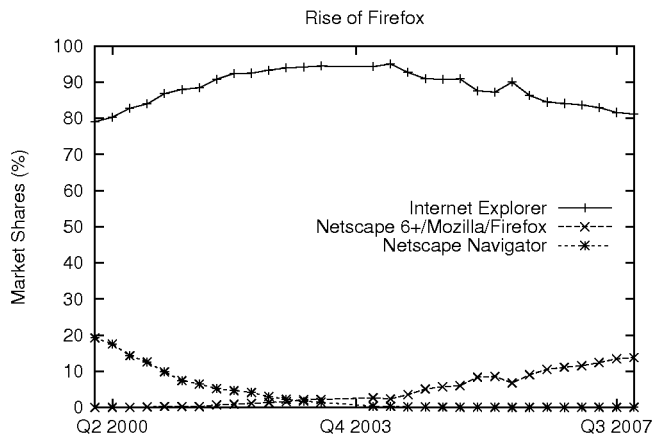
Le projet Mozilla présente enfin une structure modulaire. Les technologies Mozilla peuvent être utilisées par d'autres logiciels et bénéficier de contributions extérieures supplémentaires. Le moteur de rendu HTML, baptisé Gecko, a ainsi servi de base à d'autres navigateurs Web comme Camino ou Galeon. Avec le recul, il apparaît cependant que les outils de développement mis au point pour le projet Mozilla ont peiné à se diffuser au-delà des projets de la Fondation. C'est notamment vrai pour le framework applicatif XULRunner lancé en 2007¹. XULRunner devait apporter un environnement d'exécution indépendant du système d'exploitation pour les applications basées sur XUL, un langage de description d'interfaces graphiques fondé sur XML créé dans le cadre du projet Mozilla. Le manque d'environnement de développement dédié à ces outils, les lacunes de la documentation et la montée en puissance progressive des frameworks Ajax contribuent à expliquer ce résultat en demi-teinte.

La plupart des défauts des premières versions de Mozilla ont été résolus. Le logiciel est stable et les modules tiers sont enfin disponibles. Pour répondre aux critiques de lourdeur, Mozilla est découpé en applications plus légères. Le navigateur Mozilla Firebird (rebaptisé Firefox par la suite) et le client mail Mozilla Thunderbird voient le jour en 2004.

Signe de réussite, les parts de marché repartent progressivement à la hausse. Des récompenses sont offertes au jeune projet. SSC Publications, éditeur du *Linux Journal*, publie par exemple en juillet 2003 une liste des lauréats des *Linux Journal's Editors' Choice Awards*, récompensant les meilleurs produits associés au système d'exploitation *open source* GNU/Linux, et voit dans Mozilla l'un des dix produits Linux qui comptent.

Le succès revient cependant surtout en 2004 avec Firefox. Cette percée de Firefox relance la guerre des navigateurs, gagnée par Microsoft quelques années plus tôt. Plusieurs éléments d'ordre technique ont contribué au succès du navigateur Firefox. Nous pouvons citer la protection

1. Brent STEARN, « XULRunner : A new approach for developing rich internet applications », dans : *IEEE* 11.3 (2007), p. 67–73.



Évolution des parts de marché (Internet Explorer, Netscape/Mozilla/Firefox, Netscape Navigator) de 2000 à 2007 (données : TheCounter.com, Wikipédia)

contre les popups, la bonne sécurité du logiciel (Internet Explorer est alors régulièrement critiqué sur ce plan), le système d'onglets, le respect des standards, et la possibilité de créer des extensions.

Les extensions (addons.mozilla.org) sont considérées comme un avantage compétitif important et mobilisent une large communauté de développeurs. Elles permettent de satisfaire des besoins très spécifiques, donc d'attirer aussi de nouveaux utilisateurs¹. En juillet 2012, plus de 85% des utilisateurs de Firefox utilisaient au moins une extension, pour une moyenne de 5 extensions installées par utilisateur. Ces extensions couvrent des fonctionnalités aussi diverses que le blocage des publicités, le téléchargement de vidéos, la personnalisation de l'interface du navigateur ou le débogage des pages Web. La Fondation Mozilla revendique en outre plus de vingt-cinq mille développeurs d'extensions et trois milliards d'extensions téléchargées dans le monde.

Devenue autonome, la Fondation Mozilla a dû adapter son organisation, prendre en charge de nouvelles tâches (notamment le marketing) et trouver un modèle d'affaires permettant de soutenir son activité.

1. KRISHNAMURTHY, *op. cit.*

D'un point de vue organisationnel, des bouleversements se sont produits suite au désengagement d'AOL, bien que cela soit peu visible pour un observateur extérieur au projet. La transition s'est accompagnée d'une baisse sensible d'activité des développeurs et par une modification importante de la composition du groupe des développeurs les plus actifs¹. Globalement, le projet Mozilla conserve une structure de décision centralisée. Le projet dans son ensemble est géré par un groupe réduit de développeurs². Ce groupe restreint délègue un certain nombre de tâches et de rôles. Ils désignent en particulier les propriétaires de module, chargés de sélectionner et de mettre en œuvre les changements.

Les utilisateurs peuvent pour leur part signaler les bugs et discuter de leur résolution. En pratique, le développement de Mozilla est *bug-driven*, c'est-à-dire conduit par les bugs³. Le terme bug recouvre à la fois les défauts, les améliorations et les changements de fonctionnalités. Les rapports de bugs sont encodés et numérotés, puis triés. S'ils sont confirmés, leur correction est planifiée en fonction de la gravité et de la priorité. Un responsable est désigné pour chaque bug. Les propositions de modifications du code source sont finalement attachées au bug sous forme de *patches*. Ces opérations sont gérées grâce au logiciel Bugzilla, un système de suivi des problèmes basé Web développé en Perl pour les besoins du projet Mozilla.

L'organisation hiérarchisée du projet Mozilla est la source de fréquentes disputes mais permet d'assurer la cohérence du développement. Cette structure est peut-être aussi un reliquat de la forte implication passée d'une entreprise commerciale dans le développement du logiciel. Le projet Mozilla se distingue cependant par l'importance des contributions proposées

1. Jesus M. GONZALEZ-BARAHONA, Gregorio ROBLES et Israel HERRAIZ, « Impact of the Creation of the Mozilla Foundation in the Activity of Developers », dans : *Proceedings of the Fourth International Workshop on Mining Software Repositories*, 2007.

2. Sandeep KRISHNAMURTHY, « About Closed-door Free/Libre/Open Source (FLOSS) Projects : Lessons from the Mozilla Firefox Developer Recruitment Approach », dans : *Upgrade 6.3* (2005), p. 28–32. Audris MOCKUS, Roy T. FIELDING et James D. HERBSLEB, « Two case studies of open source software development : Apache and Mozilla », dans : *ACM Trans. Softw. Eng. Methodol.* 11.3 (2002), 309–346.

3. REIS et MATTOS FORTES, *op. cit.*

par les membres périphériques, ce qui tendrait à démontrer la pertinence de ce mode d'organisation pour un projet de cette envergure ¹.

D'un point de vue marketing, la Fondation Mozilla a rapidement modifié son ancien site, qui contenait quelques défauts majeurs, en particulier d'un point de vue « commercial ». Il se présentait davantage comme un site pour développeurs, avec une trop grande complexité pour le simple utilisateur. Pour ce dernier, l'information y était à la fois trop abondante et mal structurée. Par exemple, une multitude de versions de la suite Mozilla étaient proposées en bas de page. Ou encore il pouvait arriver qu'une version Bêta éclipse une version stable. Or, un simple utilisateur ne sait pas forcément ce qu'implique une version Bêta (il s'agit de la deuxième période d'essai d'un produit informatique avant sa publication officielle). Les choses ont positivement changé lors de la sortie de la version 1.4 du navigateur (mi-2003). La dernière version stable était clairement mise en évidence (coin supérieur gauche de la page). Les fichiers d'installation étaient classés par système d'exploitation (Linux, Mac, Windows). L'information recherchée par le simple utilisateur était donc accessible directement. Mozilla a également introduit la notion de « produit ». Familière aux utilisateurs de logiciels, elle implique la considération du client et un minimum de réflexion stratégique. On peut même ici parler de gamme de produits, puisque l'utilisateur pouvait notamment choisir entre le classique Mozilla et une nouvelle version allégée du navigateur, Firebird (devenue Firefox par la suite). Ce développement d'une gamme de produits allait de pair avec l'élaboration d'une stratégie de marques (Mozilla Application Suite, Thunderbird, Firebird, etc.). À noter que le nom Firefox est déposé comme marque. Des références (prix remportés auprès de revues, citations de personnes reconnues) étaient clairement mises en évidence, de façon à crédibiliser les produits et à rassurer les nouveaux utilisateurs attirés sur le site. Les contributeurs n'avaient pas été négligés puisque, plus bas sur la page, des informations à destination des testeurs et développeurs apparaissaient dans une couleur différente. Les systèmes d'exploitation supportés et l'outillage mis à disposition étaient également mis en avant. Les nouvelles initiatives de Mozilla, comme l'assistance

1. Jing WANG et J. M. CARROLL, « Behind Linus's law : A preliminary analysis of open source software peer review practices in Mozilla and Python », dans : *2011 International Conference on Collaboration Technologies and Systems (CTS)*, 2011, p. 117-124.

téléphonique, la vente de cédéroms et les dons, étaient pointées sur l'ensemble du site.

La Fondation Mozilla a ultérieurement poursuivi ses efforts marketing. Ainsi, après avoir testé la publicité classique avec l'achat d'une page de publicité (financée par la communauté) dans la presse américaine, la Fondation Mozilla a testé en 2005 le marketing viral. Le site FunnyFox (www.funnyfox.fr) a ainsi hébergé trois spots publicitaires humoristiques réalisés par l'agence française de communication Pozz : *The Office*, *The Mobile* et *The Notebook*. L'espoir était que ces trois films en Flash soient largement diffusés à l'initiative des internautes et se répandent ainsi sur Internet. La Fondation est ensuite allée plus loin en matière de co-création. Cette technique marketing visant à impliquer clients et utilisateurs dans la conception, le développement, la promotion ou le support de produits et services nouveaux a été appliquée pour la création d'annonces publicitaires. Mozilla a ainsi lancé fin 2005 un concours baptisé Firefox Flicks¹. L'objectif était de créer « une campagne de marketing puisant dans l'énergie créative des réalisateurs de films amateurs et professionnels afin de présenter Firefox à un public encore plus large ».

La communauté a par ailleurs été mise à contribution. Krishnamurthy² identifie notamment trois missions prises en charge par la communauté : le développement de la marque, la création de trafic et la conversion de nouveaux utilisateurs (fourniture de bannières, écriture d'avis positifs, votes sur des sites dédiés aux logiciels, etc.).

D'un point de vue financier, la Fondation Mozilla s'est assurée des rentrées d'argent régulières par des accords avec les moteurs de recherche commerciaux. Google compte pour 95% de ses revenus. La Fondation Mozilla a annoncé avoir gagné 52,9 millions de dollars sur l'année 2005 avec son navigateur. À titre de comparaison, ses revenus s'élevaient à 2,4 millions de dollars en 2003 et 5,8 millions en 2004. Employant 70 permanents, la fondation doit sa survie financière à cet accord qui la rend dépendante de Google. Les dépenses de Mozilla Foundation et de Mozilla Corporation (filiale commerciale) pour l'année 2005 s'élèvent pour leur part à 8,2 millions de dollars. Le développement logiciel y compte pour

1. www.firefoxflicks.com.

2. KRISHNAMURTHY, « Mozilla vs. Godzilla – The Launch of the Mozilla Firefox Browser ».

1993	Sortie de NCSA Mosaic.
1994	Création de la Netscape Communications Corporation. Sortie de Netscape Navigator.
1995	Sortie de Microsoft Internet Explorer.
1998	Accord AOL – Microsoft. Libération de Netscape (projet Mozilla).
1999	Rachat de Netscape par AOL.
2000	Sortie de Netscape 6.0 basé sur Mozilla.
2003	Fin de Netscape. Autonomisation de la Fondation Mozilla.
2004	Sortie du navigateur Firefox.
2008	Sortie du navigateur Google Chrome basé sur Webkit.
2011	Lancement du projet Boot for Gecko.

Projet Mozilla : grandes dates

6 075 474 dollars, contre 768 701 dollars pour les dépenses marketing et 1 328 353 dollars pour les dépenses administratives générales.

4. 2008 – 2012 : la concurrence de Webkit

L'année 2008 a peut-être marqué un nouveau tournant dans la guerre des navigateurs. Septembre 2008 a en effet vu le lancement du navigateur Google Chrome. L'enjeu porte ici sur l'expérience des utilisateurs d'applications Internet riches¹. Ces dernières sont des applications s'exécutant dans un navigateur Web mais offrant des caractéristiques comparables à des applications exécutées sur poste de travail. Google propose ce type de produits (Youtube, GMail, etc.) et affirme avoir été confronté lors de tests à des problèmes de stabilité et de rapidité des navigateurs existants. Google a dès lors décidé de sortir son propre navigateur.

Google Chrome est un navigateur basé sur Webkit (webkit.org). Webkit est un projet *open source* lancé en 2005 par Apple pour son navigateur Safari. Il se base sur le code source de KHTML, le moteur de rendu HTML

1. J. Ken CORLEY et D. Scott HUNSIGER, *Why are People Using Google's Chrome Browser ?*, Wilmington : CONISAR Proceedings, 2011.

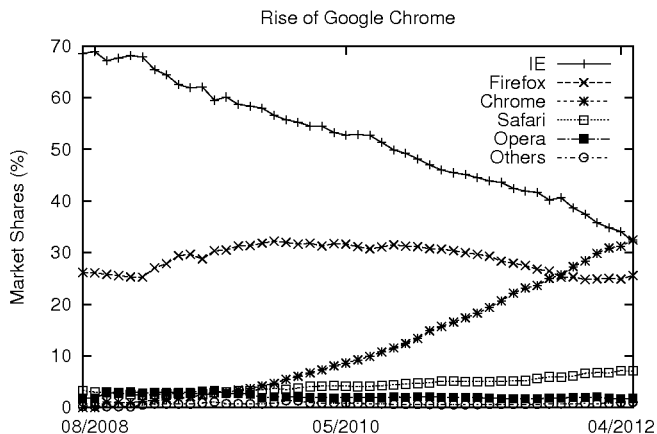
utilisé par Konqueror, le gestionnaire de fichiers et navigateur Web associé à l'environnement de bureau KDE. Apple a été rejoint sur ce projet par Nokia, Palm ou encore RIM (en pratique, les navigateurs basés sur Webkit dominant sur les plates-formes mobiles). Chrome possède une base libre connue sous le nom de Chromium. L'écosystème Mozilla, notamment alimenté par les entreprises souhaitant disposer d'un navigateur fonctionnel sur d'autres plates-formes que Microsoft Windows, est donc aujourd'hui concurrencé par un nouvel écosystème principalement composé d'entreprises parfois frontalement en concurrence sur certains marchés (Google et Apple, par exemple).

À sa sortie, Google Chrome s'est révélé rapide et stable (processus d'isolation). Google Chrome propose fréquemment de nouvelles versions et voit son parc rapidement renouvelé par le biais d'un système de mise à jour automatique. Le nouveau produit a été vigoureusement promotionné par Google, à l'aide notamment de spots vidéos. Il est rapidement devenu le troisième navigateur le plus utilisé, derrière Internet Explorer et Firefox. En pratique, l'évolution des parts de marchés des navigateurs montre une stabilisation de Firefox, une diminution d'Internet Explorer et une croissance de Chrome. En mai 2012, Chrome pointait à 32,43% de parts de marché, contre 32,12% pour Internet Explorer et 25,55% pour Firefox. La concurrence de Chrome semble dès lors s'exercer davantage sur le concurrent commun Internet Explorer.

Le ralentissement de l'adoption de Firefox s'est par ailleurs accompagné de tensions au sein de la communauté¹. La discorde porte principalement sur les relations entre contributeurs membres des communautés et les structures officielles du projet Mozilla. Le cycle de publication rapide inauguré par Firefox en 2011 dans le sillage de Chrome (ce cycle est notamment intéressant sur un plan marketing) a également fait l'objet de discussions passionnées. Le bilan de cette nouvelle politique ne semble cependant pas négatif, avec une stabilité globalement préservée et une correction plus rapide des erreurs)².

1. Guillaume CHAMPEAU, « Mozilla interroge sur son avenir et ménage sa communauté », dans : *Numérama* (2011)

2. Foutse KHOMH et al., « Do faster releases improve software quality? An empirical case study of Mozilla Firefox », dans : *9th IEEE Working Conference on Mining Software Repositories (MSR)*, 2012, p. 179–188.



Évolution des parts de marché (Internet Explorer, Chrome, Firefox) de 2008 à 2012 (données : Statcounter.com).

Cette nouvelle soulève des questions quant à l'avenir de l'accord liant actuellement la Fondation Mozilla et Google, ainsi qu'à la pérennité financière de la Fondation. À plus long terme, les motivations derrière la sortie de Chrome interrogent aussi sur l'évolution du Web voulue par les grands groupes commerciaux. La frontière entre Internet et le poste de travail tend ainsi progressivement à s'estomper, de même que celle entre page Web et application.

La concurrence de Google Chrome / Chromium va par ailleurs de pair avec celle, plus globale, du moteur de rendu Webkit (face à Gecko), en particulier dans les systèmes mobiles. La plupart des navigateurs mobiles utilisent ainsi Webkit. Mozilla se retrouve marginalisé sur ces plateformes, alors qu'elles deviennent proportionnellement de plus en plus importantes. L'utilisation de tablettes et de smartphones deviendrait déjà, par exemple, dominante pour la consultation des sites d'actualités pendant certaines périodes de la journée (source comScore). Mozilla tente de se repositionner sur ce marché avec le projet Boot to Gecko, un système d'exploitation ouvert pour terminaux mobiles.

5. Discussion

Le succès initial de Netscape a été analysé par Cusumano et Yoffie¹, qui l'attribuent à l'adoption d'une pratique de développement flexible favorisant l'interactivité avec les utilisateurs et l'évolution des fonctionnalités en fonction des retours utilisateurs obtenus. Deux éléments supplémentaires nous semblent cependant pouvoir être mentionnés.

Le premier est la filiation de Netscape Navigator avec NCSA Mosaic. Netscape bénéficiait en effet de l'expertise de pionniers de l'Internet et démarrait sur des concepts techniques maîtrisés. Cette situation peut être comparée à celle du logiciel libre Apache HTTPd, un serveur Web libre dominant depuis le milieu des années 1990, lui-même basé sur le code source du serveur Web publié par le NCSA.

Le second est l'intérêt bien compris par Microsoft d'exploiter ses ressources internes pour améliorer le produit de manière itérative (les fonctionnalités sont ajoutées par incréments testés quotidiennement). Lorsque l'on parle de co-création, on entend généralement une collaboration entre l'entreprise et ses clients ou utilisateurs dans le but de créer une valeur supplémentaire partagée. Il serait cependant dommage de négliger le potentiel représenté par le personnel de l'entreprise, surtout dès lors que celle-ci dispose d'une main-d'œuvre abondante. Microsoft valide de la sorte en interne divers scénarios d'utilisation des technologies qu'il édite. Ce principe du *dogfooding* reste d'application chez Microsoft. Face à une entreprise de grande taille, le recours aux clients dans le cadre d'une initiative de co-création ne semble en tout cas pas toujours suffire à maintenir le rythme d'innovation nécessaire à une différenciation sur le long terme.

La pratique du développement itératif pose également la question de l'impact sur le design de l'application. L'érosion du design peut en effet conduire à un re-développement complet d'un logiciel ou de certaines de ses parties. Ce travail a dû être entrepris pour le logiciel Netscape lors de sa libération en 1998². L'effet positif des méthodes de développement itératif doit donc être nuancé.

Les débuts difficiles du projet Mozilla suggèrent que le passage d'un mode de développement propriétaire à un mode de développement libre

1. CUSUMANO et YOFFIE, *op. cit.*

2. GURP et BOSCH, *op. cit.*

et communautaire nécessite une préparation importante. Cet effort supplémentaire inclut la mise au point d'outils de travail (ex. : Bugzilla) et un *reengineering* du code (ex. : Gecko). Ce dernier comprend une clarification de la syntaxe et une modularisation du code permettant le passage à un mode de travail collaboratif. Ce constat tempère la recommandation d'Eric Raymond, souvent reprise, voulant que le développeur de logiciel libre doit diffuser tôt et souvent. Dans le même ordre d'idée, Bar et Fogel¹ estiment que la courbe d'apprentissage de l'utilisateur du code source doit être rapide. En particulier, ils en déduisent que l'architecture du logiciel doit être compréhensible et réfléchie de manière à faciliter la maintenance et l'ajout de fonctionnalités. L'importance de la modularité et d'une faible complexité du code source a d'ailleurs été confirmée comme facteur de succès pour les projets *open source*².

La mise en place d'un système d'interface de programmation (API) simplifiant la création d'extensions est un moyen pour faciliter les contributions extérieures, qui ne nécessitent dès lors plus une connaissance intime des fondements du logiciel. Il s'agit justement d'un avantage compétitif reconnu de Firefox à sa sortie.

Si le projet communautaire en lui-même a progressivement su trouver son souffle et proposer plusieurs innovations (navigation par onglet, système d'extensions, respect des standards W3C...), il n'en va pas de même de la version commerciale du navigateur, combinant le logiciel libre Mozilla et différentes extensions. Le logiciel communautaire est en effet devenu en quelques années davantage populaire que son dérivé commercial. Cependant, la popularité du logiciel est finalement allée de pair avec un réel travail marketing conduisant à la création d'une véritable ligne de produits et à une présentation plus attractive des produits et services offerts. La complémentarité entre la communauté et une structure agissant pour sa promotion (association, fondation, entreprise privée...) apparaît ici clairement.

La difficulté à accorder des rythmes de travail différents ressort comme un défi important. La « querelle autour d'une roadmap » illustrée par la phrase « vos besoins ne sont pas nos besoins pour l'instant » est un risque

1. Moshe BAR et Karl FOGEL, *Open Source Development With Cvs*, 3rd Revised edition, Scottsdale : Paraglyph Inc., 2003.

2. COMINO, MANENTI et PARISI, *op. cit.*

(cité par François Elie ¹) en matière de collaboration entre une entreprise et une communauté de logiciel libre. Là où une communauté peut continuer à son rythme sans être trop pressée par des impératifs de rentabilité, une structure commerciale doit s'en tenir à une planification plus stricte et réduire ses délais de mise sur le marché. La préparation technique, organisationnelle et financière de la publication sous licence libre n'en est que plus importante.

Les choix opérés pour le projet Mozilla recourent par ailleurs les recommandations faites par plusieurs études sur les facteurs de succès de projets *open source* ². Par exemple, l'impact négatif des licences dites restrictives sur le succès des projets *open source* tend à conforter le choix d'une nouvelle licence apportant un compromis équilibré entre les aspirations des entreprises commerciales et celles des contributeurs bénévoles. Par ailleurs, la création de grandes communautés s'accompagne de structures organisationnelles et de gouvernance plus complexes. La mise en place d'une structure plus formelle peut contribuer à générer des tensions internes telles qu'on peut occasionnellement l'observer au sein de la communauté Mozilla. La présence structurante d'un sponsor, institution à but non lucratif ou entreprise, est également reconnue comme facteur de succès.

La relation entre Google et Mozilla illustre l'évolution des relations existant entre entités au sein d'un écosystème commercial. Google est ainsi à la fois un soutien financier (au travers d'un accord publicitaire) et un concurrent (au travers des logiciels Chrome et Chromium) de la Fondation Mozilla. Cette situation met en évidence la relative fragilité de la structure (même si ses besoins financiers sont assez modestes en comparaison de ses revenus) qui soutient le développement de Firefox puisque sa principale source de revenus est *de facto* un de ses concurrents.

Un nouveau champ de bataille s'est par ailleurs ouvert ces dernières années, avec le développement du Web mobile. Il dépasse largement la seule problématique des navigateurs et rappelle, par ses enjeux (en particulier : le respect des standards), la première guerre des navigateurs. Les plates-formes iOS (iPhone, iPad et iPod touch) et Android ont ainsi popularisé le

1. François ÉLIE, *Économie du logiciel libre*, Paris : Eyrolles, 2009.

2. COMINO, MANENTI et PARISI, *op. cit.*

concept d'*Apps*¹. Ces dernières sont des applications natives spécifiques à une plate-forme mobile donnée. Elles apportent un *look & feel* propre (c'est-à-dire une apparence et des commandes spécifiques) et permettent une intégration plus étroite avec la plate-forme. Elles échappent cependant en partie aux efforts de standardisation pour l'accès aux contenus et applications en ligne. Les développeurs se trouvent donc face à une alternative : proposer des *Apps* propres à chaque plate-forme ou rester dans la ligne du Web ouvert en se basant sur des standards tels que l'HTML5, CSS, Javascript ou WebGL. Les critères de choix sont autant d'ordre technique (accès à des fonctionnalités avancées, évolution parfois plus lente des standards, etc.) que financier (facilité de monétisation, coûts de développement, etc.). Lourdemment concurrencé par Webkit sur iOS et Android, contré par Google Chrome (également basé sur Webkit) sur poste de travail, Mozilla tente aujourd'hui de se repositionner sur le marché mobile avec Boot to Gecko.

Conclusion

Trois éléments nous semblent avoir été déterminants dans l'échec du navigateur Netscape.

Premièrement, Netscape a surestimé sa position sur le marché, se jugeant incontournable du fait de sa supériorité technique. L'entreprise s'est montrée incapable de s'adapter aux évolutions de sa situation concurrentielle et d'élaborer une stratégie à long terme lui permettant de pérenniser son activité. Lors du rachat par AOL, le navigateur était déjà un produit en déclin commercial.

Deuxièmement, la différence de taille entre Netscape et Microsoft donnait à cette guerre des navigateurs un côté David contre Goliath. Les capacités d'investissement de Microsoft lui ont permis de se positionner rapidement sur le marché. Son expérience de la négociation commerciale ainsi que la puissance de ses réseaux de distribution (la politique commerciale de Microsoft a par ailleurs fait l'objet d'une longue et médiatique procédure judiciaire) l'ont aussi favorisé dans les tractations avec AOL.

1. Tommi MIKKONEN et Antero TAIVALSAARI, « Apps vs. Open Web : The Battle of the Decade », dans : *Proceedings of 2nd Workshop Software Eng. for Mobile Application Development*, 2011.

Troisièmement, le manque de préparation du projet Mozilla a entraîné des retards dommageables à Netscape dans un marché en évolution très rapide. Il faut cependant souligner le manque d'expérience en la matière en 1998, et le rôle pionnier que le projet Mozilla a joué.

Trois éléments nous semblent avoir été déterminants dans le succès du navigateur Firefox.

Premièrement, la jeune Fondation Mozilla s'est montrée à l'écoute des demandes des utilisateurs et a su tirer parti des faiblesses de son concurrent. Plus léger, sécurisé et modulaire, Firefox améliorerait sensiblement l'expérience des utilisateurs (et des développeurs Web).

Deuxièmement, la Fondation a mis en place des partenariats assurant des rentrées financières lui permettant de rémunérer une partie des développeurs et d'assurer l'organisation du projet.

Troisièmement, bien que réalisé dans un cadre non commercial et impliquant des communautés de contributeurs bénévoles, les responsables du projet ont développé une politique marketing visant à acquérir de nouveaux utilisateurs. Il s'agit d'une démarche originale dans ce contexte.

Le projet se trouve aujourd'hui confronté à un enjeu majeur, lié à l'évolution du marché et à l'adaptation face aux nouveaux concurrents. Le Web évolue progressivement sous la pression des entreprises d'un monde de documents, élaborés à l'aide de standards, vers un monde d'applications, plus lourdes et complexes. La sortie de Google Chrome, basé sur Webkit, s'inscrit dans ce contexte. Autre évolution : l'essor du Web mobile. Si la situation sur poste de travail est actuellement concurrentielle, avec un partage équilibré du marché entre Internet Explorer, Firefox et Chrome, le marché des systèmes mobiles a par contre évolué vers une situation de monopole au profit du moteur de rendu Webkit, sur lequel se basent la plupart des navigateurs mobiles. Alors que les systèmes d'exploitation mobiles disposent d'un navigateur préinstallé, le succès de Boot to Gecko sera vraisemblablement déterminant pour l'avenir de la Fondation, compte tenu de l'importance croissante du Web mobile...

Remerciements

Une première version de cette étude a fait l'objet d'une présentation dans le cadre du Colloque de l'Association Information et Management

en 2011. Nous remercions les relecteurs pour les pistes d'améliorations suggérées lors de la soumission de l'article.

Références

- BAR, Moshe et Karl FOGEL, *Open Source Development With Cvs*, 3rd Revised edition, Scottsdale : Paraglyph Inc., 2003.
- BERNERS-LEE, Tim et al., « The World-Wide Web », dans : *Communications of the ACM* 37.8 (1994), p. 76–82.
- CERUZZI, Paul E., « Aux origines américaines de l'Internet : projets militaires, intérêts commerciaux, désirs de communauté », dans : *Le Temps des Médias* 18.1 (2012), p. 15, URL : <http://www.cairn.info/resume.php?IDARTICLE=TDM0180015>.
- CHAMPEAU, Guillaume, « Mozilla interroge sur son avenir et ménage sa communauté », dans : *Numérama* (2011).
- COMINO, Stefano, Fabio Maria MANENTI et Maria Laura PARISI, « From planning to mature : On the success of open source projects », dans : *Research Policy* 36.10 (2007), p. 1575–1586.
- COOL, Yorick et al., *Les logiciels libres face au droit*, Bruxelles : Cahier du CRID, Bruylant, 2005.
- CORLEY, J. Ken et D. Scott HUNSIGER, *Why are People Using Google's Chrome Browser ?*, Wilmington : CONISAR Proceedings, 2011.
- CUSUMANO, Michael A. et David B. YOFFIE, *Competing On Internet Time : Lessons From Netscape And Its Battle With Microsoft*, New York : Touchstone, 2000.
- DESCOMBES, Serge, « Saga Netscape/Microsoft : histoire d'un renversement », dans : *Journal du Net* (2002).
- DI PENTA, Massimiliano et Daniel M. GERMAN, « An Exploratory Study of the Evolution of Software Licensing », dans : *ICSE'10* Mai (2010).
- ÉLIE, François, *Économie du logiciel libre*, Paris : Eyrolles, 2009.
- GONZALEZ-BARAHONA, Jesus M., Gregorio ROBLES et Israel HERRAIZ, « Impact of the Creation of the Mozilla Foundation in the Activity of Developers », dans : *Proceedings of the Fourth International Workshop on Mining Software Repositories*, 2007.
- GURP, Jilles van et Jan BOSCH, « Design erosion : problems and causes », dans : *Journal of System and Software* 61.2 (2002), 105–119.

- HALLOUL, Rached, « Le réseau stratégique et la concurrence illustrés par le cas M/N (Microsoft versus Netscape) », dans : *Innovations* 21 (2005), p. 197–216.
- HAMERLY, Jim et Tom PAQUIN, « Freeing the Source : The Story of Mozilla », dans : *Open Sources : Voices from the Open Source Revolution*, sous la dir. de Chris DiBONA, Sam OCKMAN et Marc STONE, Sebastopol : O'Reilly, 1999, p. 197–206, URL : <http://oreilly.com/openbook/opensources/book>.
- HARRISON, Warren, « Eating your Own Dog Food », dans : *IEEE Software* 23.3 (2006), p. 5–7.
- IANSITI, Marco et Alan MACCORMACK, « Developing Products on Internet Time », dans : *Harvard Business Review* 75.5 (1997), p. 108–117, URL : <http://hbr.org/1997/09/developing-products-on-internet-time/sb1>.
- KHOMH, Foutse et al., « Do faster releases improve software quality ? An empirical case study of Mozilla Firefox », dans : *9th IEEE Working Conference on Mining Software Repositories (MSR)*, 2012, p. 179–188.
- KRISHNAMURTHY, Sandeep, « About Closed-door Free/Libre/Open Source (FLOSS) Projects : Lessons from the Mozilla Firefox Developer Recruitment Approach », dans : *Upgrade* 6.3 (2005), p. 28–32.
- « Mozilla vs. Godzilla – The Launch of the Mozilla Firefox Browser », dans : *Journal of Interactive Marketing* 23.3 (2009), p. 259–271.
- LAAT, Paul B. de, « Copyright or copyleft ? An analysis of property regimes for software development », dans : *Research Policy* 34 (2005), p. 1511–1532.
- LIOTARD, Isabelle, « Les nouvelles facettes de la propriété intellectuelle : stratégies, attaques et menaces », dans : *Management et Sciences Sociales* 4 (2007), p. 1–14.
- MIKKONEN, Tommi et Antero TAIVALSAARI, « Apps vs. Open Web : The Battle of the Decade », dans : *Proceedings of 2nd Workshop Software Eng. for Mobile Application Development*, 2011.
- MOCKUS, Audris, Roy T. FIELDING et James D. HERBSLEB, « Two case studies of open source software development : Apache and Mozilla », dans : *ACM Trans. Softw. Eng. Methodol.* 11.3 (2002), 309–346.

- REIS, Christian Robottom et Renata Pontin de MATTOS FORTES, *An Overview of the Software Engineering Process and Tools in the Mozilla Project*, Working paper, 2002.
- SEBENIUS, James K., « Negotiating Lessons from the Browser Wars », dans : *MIT Sloan Management Review* (2002).
- STEARNS, Brent, « XULRunner : A new approach for developing rich internet applications », dans : *IEEE 11.3* (2007), p. 67–73.
- WANG, Jing et J. M. CARROLL, « Behind Linus's law : A preliminary analysis of open source software peer review practices in Mozilla and Python », dans : *2011 International Conference on Collaboration Technologies and Systems (CTS)*, 2011, p. 117–124.
- WANG, Tongsen, Lei WU et Zhangxi LIN, « The revival of Mozilla in the browser war against Internet Explorer », dans : *Proceedings of the 7th international conference on Electronic commerce*, New York : ACM, 2005, 159–166.
- ZAWINSKI, Jamie, *Resignation and postmortem*, www.jwz.org, 1999.

Pure data, logiciel libre de productions créatives

Séverine GIORDAN

« Pd » acronyme de « Pure Data », encore dénommé « générateur de codes créatifs » ou « mécano cybernétique »¹ est présenté aujourd'hui, et dans un sens général, comme un logiciel à programmation graphique pour les créations musicales, multimédias en temps réel. Toutefois, *Pure Data* s'affranchit largement de cette définition générique. *Pure Data* est un logiciel libre. Le programme informatique du logiciel est consultable, manipulable par tous. Cette caractéristique a étoffé ses fonctions, redirigé ses pistes de recherche et rendu possibles de nouvelles méthodes, de nouvelles problématiques, de nouvelles applications, provenant des pratiques des utilisateurs.

Nous chercherons ici à cerner le coefficient utopique de ce logiciel libre : initie-t-il des actions collectives, des mutualisations, des contributions, des relations équilibrées entre concepteurs et utilisateurs constituées à partir des usages ?

Où se situe *Pure Data* dans le paysage contrasté des logiciels libres ? Dans un premier temps, l'histoire du logiciel permettra de comprendre

1. Voir Roland CAHEN, présentation Jérôme ABEL http://impala.utopia.free.fr/pd/patches/PureData_Initiation_fr.pdf.

son évolution. Puis nous verrons comment des communautés autour du logiciel *Pure Data* ont émergé. Comment se forment-elles ? Quels sont leurs dynamiques et leur état d'esprit ? Enfin, les atouts de ce logiciel libre seront révélés. Ils éclairent les processus d'émergence et font entrevoir de nouvelles formes de coopérations personnelles ou institutionnelles, voire de solidarités. La diffusion et l'apprentissage de ce logiciel montrent comment de nouvelles économies, des industries créatives même, peuvent apparaître.

1. *Pure Data*, un logiciel de programmation visuelle

Pure Data est à l'origine un logiciel de programmation graphique sur un mode visuel ou auditif¹. Un *patch*, morceau de code qui conduit à la création d'un comportement en temps réel est créé par un ensemble de boîtes² (fonctions) liées les unes aux autres. Ce langage de programmation non linéaire rappelle les studios électroniques ou encore les synthétiseurs modulaires : « le résultat d'un calcul que vous avez effectué quelque part sera dirigé au moyen d'une connexion vers un module quelconque qui lui-même sera redirigé, vers un autre module etc. »³. De la sorte, les signaux électriques produits se déplacent de module en module.

Le graphisme du logiciel est une évocation des programmes modulaires à l'origine des premiers synthétiseurs modulaires analogiques, inventés par Max Mathews, premier directeur scientifique de l'IRCAM à la fin des années 1970. Il s'agit d'un programme orienté objet à la différence de la programmation procédurale plus usuelle.

Le « programme orienté objet » fonctionne sur un système modulaire. Ce parti-pris favorise l'autonomie de la pluralité des possibilités du logiciel. L'interface graphique de programmation a été pensée par rapport aux utilisateurs qui ne se sont pas forcément initiés à la programmation informatique, et ne les oblige donc pas à coder. Ce type de programmation rend abordable, accessible, ludique et appropriable un domaine de

1. François PACHET et Jean-Pierre BRIOT, eds., *Informatique musicale : Du signal au signal musical*, Paris : Hermès, 2004, p. 21.

2. Des « objets », qui sont l'instance de classe, c'est-à-dire l'instance de données et de manipulateurs de données.

3. Voir Archives IRCAM, *Manuel Max*, p. 1.

compétences scientifiques et technologiques qui peut sembler hermétique, lorsqu'il est observé du point de vue de l'apprentissage du langage informatique.

2. Dialogue entre scientifiques et artistes

Pure Data est issu de différents projets. Ils se nomment respectivement *4X*, *Pattern Matching*, *Patcher*, *Max*, *Max/MSP* ou *JMax*... Lors de sa mise au point, il a « voyagé » dans les plus grandes institutions de recherche du monde. L'IRCAM¹, institut de recherches et de créations musicales associé au Centre Pompidou, établissement public à caractère culturel, fut un point de départ.

La politique de l'IRCAM et certains de ses projets permettent de comprendre l'évolution de ce logiciel. Cette institution a toujours eu un axe majeur spécifique qui est le dialogue entre la recherche scientifique et la création. Ce dialogue a pour vocation de conduire à de nouvelles formes expressives, créatives, nourrissant tout autant les inventions des musiciens que celles des scientifiques. L'IRCAM est très attentif aux liens possibles entre la culture musicale, son histoire, ses formes et les processus scientifiques et technologiques. Les résultats de ces recherches sont appropriés par des processus artistiques et conduisent alors à l'émergence de questionnements nouveaux. Pierre Boulez², directeur de l'IRCAM, reproche au GRM³ (Groupe de Recherche Musicale) d'instrumentaliser les processus artistiques à des fins scientifiques et technologiques. L'élaboration d'outils, de méthodes, de synthèses sonores, de création d'instruments de musiques, d'analyse, de traitements, de transformations du son en temps réel, des dispositifs d'acoustique sonore dans l'espace furent et restent les fruits de cette institution.

1. Institut de Recherche et de Coordination Acoustique / Musique.

2. Compositeur, chef d'orchestre. Directeur de l'IRCAM de 1977 à 1992.

3. Pierre Boulez emploie l'expression « bricolage empirique » pour qualifier les expérimentations du GRM. Voir Emmanuel GRYSZPAN, « Confluence et divergences. La techno face aux musiques savantes », dans : *Musiques, arts et technologies. Pour une approche critique*, sous la dir. de Roberto BARBANTI et al., Paris : l'Harmattan, 2004, p. 297. Le GRM, créé en 1958, est le centre de recherche musical dans le domaine du son et de l'électroacoustique au service de la recherche de radio-télévision française ORTF, INA.

L'IRCAM, afin de répondre à sa politique artistique et culturelle, a choisi dès sa création de prolonger les recherches de Luciano Berio du Studio de phonologie musicale de la RAI à Milan en enrichissant la 4A qui est un empilage d'une soixantaine de calculateurs (1976), émettant de multiples sons de manière autonome, en *4X* (1978). Cette machine donne la possibilité de varier les paramètres du son en temps réel grâce à la vélocité de calculs de centaines d'oscillateurs. La *4X* sous la direction technique de Peppino di Giugno, a permis à l'époque de jouer par exemple la pièce sonore « Repons »¹ de Pierre Boulez. La vitesse de calcul joue un rôle majeur dans l'évolution et la simulation du suivi en temps réel des comportements instrumentaux, dans l'intégration des phénomènes d'interprétation à la composition musicale et électroacoustique, de l'analyse du signal².

Miller Puckette, résident de l'IRCAM, mathématicien du MIT³, travaille en 1985 à un programme nommé *Pattern Matching* qui assure le recouvrement entre un instrument de musique et une interprétation électronique⁴ et soutient la cohérence entre les deux jeux. L'application reconnaît les timbres, les hauteurs et les harmonies du soliste, et la « machine » peut suivre ce dernier dans l'évolution de son jeu. De 1986 à 1987, il développe *Patcher*, qu'il définit en ces termes : « contrôle raffiné des synthétiseurs pour l'exécution *live* »⁵. *Patcher*, environnement graphique de programmation musicale en temps réel, est l'aboutissement du travail commun du compositeur Philippe Manoury et du mathématicien-programmeur Miller Puckette⁶, avec lequel ils créent l'œuvre musicale *Jupiter*⁷. Il s'agit d'une pièce dont l'environnement informatique (*4X*, 1986) reconnaît la flûte. *Jupiter* est la première pièce du cycle *Sonus Ex*

1. Œuvre de Pierre Boulez (1981-1982-1984) avec comme sujet le dialogue entre jeu individuel et jeu collectif avec comme formation 6 instruments solistes, un ensemble instrumental et un système électro-acoustique pour transformer et spatialiser le son des solistes. Voir <http://brahms.ircam.fr/works/work/6997>.

2. *Stereolux*, (Journées du code créatif 29-30 mars 2012) – Miller Puckette précise les moments clés des avancées technologiques pour le temps réel au début de son intervention.

3. Massachusetts Institute of Technology

4. Archives IRCAM, *Atelier de recherche Musicale*, 14.11.1985.

5. Archive IRCAM, *Manuel Max*, p. 1

6. « Miller Puckette travaille principalement à rendre le programme fiable et maniable ». IRCAM, « Philippe Manoury », dans : *Les Cahiers de l'Ircam* 8 (1995), p. 40.

7. Prix de la meilleure réalisation musicale de la SACEM en 1988.

*Machina*¹. Philippe Manoury avait suivi un enseignement de la composition musicale assistée² par ordinateur et avait débuté cette forme de composition avec l'IRCAM au début des années 1980. Dans *Jupiter*, il expérimente l'interaction instrument/machine en temps réel. À cette fin, Philippe Manoury décrit trois types de démarche de composition dans l'interaction instrument/machine :

- La programmation est réalisée antérieurement. En *live*, la transformation du rythme est envisageable, par contre les événements, le découpage du temps sont fixés.
- Le jeu peut être aussi lié à des calculs probabilistes.
- Les signaux sonores provenant d'entrées Midi³ sont interprétés par la machine.

La notion d'« interprétation » est alors questionnée à partir de ces nouveaux possibles ; elle intègre les innovations scientifiques et technologiques de Miller Puckette. Par exemple, dans la pièce *Pluton*⁴ (1988-1989), *Patcher* donne la possibilité d'étirer les sons. Philippe Manoury expérimente alors notamment les prolongements des sons du piano. La machine capte et collecte une séquence jouée par un pianiste : notes, rythmes, événements sonores, et l'ensemble de ces variables hauteurs, durées, longueurs, dynamiques sont liées à des règles⁵.

Patcher, développé au départ sur la 4X, fut ensuite amélioré sur l'ISPW⁶, la Station d'informatique musicale, une machine plus performante. *Patcher* devient alors Max⁷ en 1990. Ce n'est plus un programme spécifiquement élaboré pour une seule pièce. Il se présente comme « uni-

1. *Sonus ex machina*, réalisé à l'Ircam est composé de *Jupiter* 1987 (pour flûte), *Pluton* 1988, 1989 (pour piano), *La Partition du Ciel et de l'Enfer*, 1989 (pour orchestre) et *Neptune* 1991 (pour trois percussionnistes), confrontés à la machine et à ses différents modes de réaction.

2. *Zeitlauf* (1982) est la première pièce de Philippe Manoury réalisée à l'IRCAM. Agencement de chœurs mixtes, d'un ensemble instrumental, de synthétiseurs, de bandes magnétiques.

3. Normes d'échange de données musicales numérisées.

4. <http://brahms.ircam.fr/works/work/10493>.

5. Par exemple pour cette pièce, l'artiste introduisait les propriétés des chaînes de Markov, c'est-à-dire des développements aléatoires qui ont pour conséquence des probabilités de successions d'événements. Les traits stylistiques du soliste sont toutefois maintenus. Daniela LANGER, *Va-et-vient. Entretien avec Philippe Manoury*, Paris : Musica Falsa, 2001, p. 111.

6. IRCAM Signal Processing Workstation Stéréolux, Conférence Miller Puckette.

7. Ce nom a été donné en hommage à Max Mathews.

versel » et le *soft*¹ se conforme à toutes pièces musicales. Cependant, le passage entre les deux machines ne se fait pas sans difficulté : toutes les pièces furent recompilées². La puissance de calcul des PC rend alors possible la diffusion du logiciel hors de l'IRCAM et donne l'opportunité de commercialiser le logiciel sous une licence privée IRCAM, ce qui répond aux exigences économiques du centre de recherche³.

C'est David Zicarelli, chercheur développant la portabilité du logiciel sur PC à l'IRCAM, qui est alors responsable de l'entreprise californienne Opcode Systems, elle-même chargée de cette finalité commerciale. Parallèlement, de 1993 à 1994, il travaille à un logiciel complémentaire à Max, le MSP, un logiciel de synthèse sonore qui, d'une part, analyse et manipule le signal en temps réel sur PC et, d'autre part, comporte une bibliothèque d'objets (de fonctions) à cet effet. Miller Puckette et David Zicarelli travailleront ensemble pour créer des passerelles entre les deux programmes.

En 1995 sont élaborés Max/FTS, Max NeXT/X par François Déchelle et Maurizio De Cecco, deux nouvelles versions adaptées à la ISPW/NeXT/X, l'ordinateur propre à l'IRCAM. En 1996, Miller Puckette, alors professeur en musique au département Art et Informatique de l'université de San Diego, participe à l'ICMC⁴ avec comme objectif de reconstituer Max. Ce projet est subventionné par le Conseil Intel Research pour le Global Music Visual⁵. Une version « augmentée » en résulte, elle concède une dimension graphique au logiciel. Cette correspondance du son à l'image était négligée par l'IRCAM.

La partie graphique a été réalisée par le chercheur Mark Danks. Il programma l'extension externe GEM (Environnement graphique pour le multimédia) qui rend possible la manipulation des flux, des fichiers vidéo, des images fixes, des fichiers 3D en temps réel. Le GEM emploie la technologie de l'Open GL, très populaire pour les animations 3D ou pour les jeux

1. Le terme *soft* provient de *software*, qui signifie logiciel en anglais.

2. Passage d'un langage informatique à un autre langage informatique.

3. Malika COMBES, *L'IRCAM fête ses 30 ans : retour sur une institution culturelle atypique dans ses rapports avec ses organismes de tutelle*, <http://musique.ehess.fr/docannexe.php?id=238> : EHESS, 2007.

4. International Computer Music Conference (ICMC) Thessaloniki en Grèce, capitale culturelle européenne en 1997.

5. <http://www.visualmusic.org>.

vidéo. L'Open GL est une bibliothèque de calculs graphiques à 2 et 3 dimensions. Les musiciens/compositeurs Vibeke Sorensen et Rang Steiner ont également participé à ce projet. La pièce musicale *Lemma*¹, qui expérimente le rapport de l'interaction instrument/machine dans un contexte multimédia, est le fruit de cette collaboration.

Le logiciel *Pure Data* est le produit de cette longue histoire. En 1998, toujours à l'IRCAM, sous la responsabilité de François Déchelle et Maurizio De Cecco, est réalisée une version libre de Max, JMax (le J de JMax correspond à l'interface Java ajoutée au logiciel), afin de permettre la portabilité du logiciel sur les systèmes d'exploitation libres.

3. Questions de licences

Miller Puckette, qui a réécrit en entier le programme Max sous le nom de *Pure Data*, a conféré une licence BSD (Berkeley software distribution license)² à son logiciel, contrairement à l'IRCAM qui avait attribué une licence privée à Max. L'Open Source Initiative (OSI)³, défendant les valeurs bénéfiques économiques du logiciel libre, considère cette licence comme adéquate. La Free Software Foundation (FSF), association qui promeut les vertus sociales et éthiques du logiciel libre, a attesté les licences BSD version 2 et 3, comme compatibles avec la licence publique générale (GPL). Cette dernière garantit les libertés fondamentales suivantes : l'emploi du logiciel pour tout usage, la possibilité de consulter le programme du logiciel et de le manipuler selon ses nécessités, la diffusion de copies, le partage de versions modifiées avec la communauté. La version 1 de la Licence BSD est rejetée par la FSF pour la clause 4, dite clause de publicité. Cette clause occasionne la notification de l'en-

1. Présenté le 27 septembre 1997 au club de Jazz Milos à Thessaloniki.

2. La licence BSD (Berkeley Software Distribution License) est une licence libre utilisée pour la distribution de logiciels. Elle permet de réutiliser tout ou une partie du logiciel sans restriction, qu'il soit intégré dans un logiciel libre ou propriétaire.

3. L'OSI prône la qualité des logiciels libres contre les logiciels propriétaires. Le logiciel libre est défendu comme source de bénéfices économiques. « *Open source is a development method for software that harnesses the power of distributed peer review and transparency of process. The promise of open source is better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in* ». *A contrario* la FSF considère le logiciel libre à partir de l'utopie éthique et sociale qu'il convoque.

semble des auteurs et organisations¹ qui ont collaboré à l'écriture des programmes, ce qui est considéré comme une publicité par FSF.

Plus exactement, *Pure Data* est sous la licence BSD amplifiée. C'est une licence BSD qui a retiré la clause 4 de sa rédaction. La redistribution et l'usage sous forme de code source ou binaire, modifié ou non, sont soumis aux conditions suivantes :

1. Le copyright, la liste des conditions et l'avertissement qui la suit doivent figurer dans le code source redistribué.
2. La documentation et/ou les fichiers fournis avec une redistribution sous forme binaire doivent faire apparaître le copyright, la présente liste des conditions et l'avertissement qui la suit.
3. Le nom de l'auteur ne saurait être utilisé dans le but de promouvoir ou de légitimer un produit dérivé de ce programme sans autorisation écrite préalable.

Il s'agit d'une licence claire, concise, simple dans sa rédaction. Elle ne limite aucun usage ; elle est permissive, contrairement à la licence GNU GPL². Elle est compatible avec des licences propriétaires. Les clauses inscrivent le droit d'auteur, la liste des conditions, l'avertissement dans la version modifiée et dans la documentation de la version du code source et l'accord écrit des auteurs et de l'organisme pour l'utilisation de leur auctorialité à des fins de cautionnement scientifique et promotionnelle.

La licence BSD n'est toutefois pas choisie par tous les contributeurs développeurs de *Pure Data*. Ce fut le cas par exemple de la librairie GEM³ (Graphics Environment for Multimedia), qui est sous licence GNU GPL. Cette librairie a permis à *Pure Data* d'être un environnement graphique de programmation pour la création multimédia en temps réel donnant libre cours à l'imagination des utilisateurs par rapport à la création et les manipulations d'objet en 3D, de vidéos, d'images, de particules grâce à l'open-GL⁴. La licence GNU GPL applique le *copyleft* c'est-à-dire une protection de la qualité de logiciel libre⁵.

1. <http://www.gnu.org/philosophy/bsd.fr.html>.

2. Rédigée par Richard Stallman et Eben Moglen, dans un premier temps.

3. Mark Danks, Chris Clepper, Cyrille Henry, IOhannes M Zmoelnig, Guenter Geiger, Daniel Heckenberg, James Tittle, Hans-Christoph Steiner et bien d'autres ont participé à sa création.

4. Conception d'applications générant des images 2D et 3D.

5. Voir sur ce point : <http://fr.wikipedia.org/wiki/Copyleft>.

Le logiciel *Pure Data* possède des bibliothèques dans lesquelles l'amateur trouve des « objets » qui lui donnent des outils de création. L'« objet » [*expr~*], réalisé par Aymeric Mansoux, Chun Lee et Olivier Laruelle, sous le nom du collectif OXA¹ qui permet la composition de sons 8 bits, a été édité sous la LAL, Licence Art Libre. Les logiciels, les programmes informatiques sont ici compris comme des œuvres de création, ce qui explique l'utilisation de cette licence² qui fut créée à partir d'une liste de diffusion (Copyleft_attitude@april.org) et lors des rencontres Copyleft Attitude³. La Licence Art Libre stipule la possibilité de diffuser, de copier, de transformer, d'exploiter gratuitement ou commercialement une œuvre à la seule condition de rendre accessible l'original, la matrice du produit dérivé. Les personnes utilisant l'objet [*expr~*] sont sous le joug de la LAL, attestée par la GNU GPL. Or, malgré les différences des licences des extensions de Pure Data, lorsque le logiciel est exécuté, seule la licence BSD apparaît.

Miller Puckette conserve le contrôle de l'expansion du noyau du logiciel, c'est-à-dire la partie qui effectue les programmes de base du logiciel. Il maîtrise également les lignes directrices fortes : la fiabilité du logiciel, l'amélioration du temps réel de celui-ci, l'enrichissement de la reconnaissance du signal de la voix et la charte graphique⁴. Andrea Mayr dans l'article « Pd as Open Source Community »⁵ explique que la légitimité aucto-

1. <http://archive.org/details/GOSUB10-004>. « [*expr~*] is the first music release of Oxa, consists of retro sounding tracks made almost entirely with the [*expr~*] object in Pure Data. No tweaking of number boxes and sliders, no clever generative algorithms, just switch the audio on and listen. The source code for this release (implemented in Pure-data) is free software distributable under the terms of the GNU General Public License (version 3 or greater). For repository, see : <http://gitorious.org/Oxa>. Copyleft : This is a free work, you can copy, distribute, and modify it under the terms of the Free Art License. Released on 3rd April 2011. »

2. « Notre souci commun était que la Licence Art Libre puisse répondre à trois conditions : qu'elle soit conforme au droit français, qu'elle soit fidèle à l'esprit du Copyleft de la General Public License et qu'elle corresponde à nos intentions artistiques de partage de la création nourrie de ce que nous savions d'un certain art contemporain. » Antoine MOREAU, « Le copyleft appliqué à la création. Une reformulation des données culturelles ? », thèse de doct., Nice : Université de Nice Sophia Antipolis, 2001, p. 478.

3. Rencontres Copyleft Attitude organisées par Antoine Moreau avec les artistes réunis autour de la revue *Allotopie*, François Deck, Antonio Gallego, Roberto Martinez et Emmanuelle Gall.

4. Cf. *Les journées du code créatif* (première édition) à Nantes, *Stéréolux*. 29-30 mars 2012.

5. Andrea MAYR, « Pd as Open Source Community », dans : *Bang, Pure Data, Berlin, Wolke Verlagsges* (2004), p. 33-44.

riale du logiciel se trouve dans les qualités de création des programmes, de documentations et de communication des développeurs, ce qui explique l'aura dominante de Miller Puckette¹. Pourtant, cette autorité non négociée a des limites : des versions différentes DesireData ou Pd-12Ork sont apparues, prétendant apporter une amélioration des programmes liés aux programmes de base. De plus, certains programmeurs contribuent désormais non plus à *Pure Data* mais à d'autres logiciels de programmation en temps réel comme SuperCollidor ou vvvv.

4. La Communauté

4.1 Conventions et dispositifs de coopération

Lors d'une résidence à l'IEM (Institut de musique acoustique et électronique) de Graz en Autriche à la fin des années 1990 avec certains étudiants et chercheurs de cette formation, Miller Puckette a lancé les « catalyseurs » de la communauté. Ils ont produit tour à tour une liste de diffusion, un wiki² dédié, un guide donnant la marche à suivre pour enrichir le logiciel³ et un IRC (Internet Relay Chat). Ces outils facilitent la coopération entre développeurs et simples amateurs de *Pure Data*.

Ces dispositifs sont toujours opérants et, avec le temps, ils rassemblent une communauté importante comportant de nombreuses nationalités. Depuis lors, d'autres forums ont émergé. Ils ont pour objectif principal de mettre en relation les utilisateurs de ce logiciel à partir d'une communauté géographique ou de langues⁴.

L'IEM à l'origine de la liste de diffusion est également à l'origine de la première *Convention* tenue à Graz, en 2004. Ces conventions comportent des conférences, des séminaires, des tables rondes, des ateliers, des *works in progress*, des expositions, des performances et des concerts. Chaque convention est spécifique ; elle est à l'image de l'équipe d'accueil. La

1. Miller Puckette est depuis 1994 le directeur associé du centre de recherche en arts et informatique de l'université de San Diego.

2. <http://puredata.info>.

3. Voir <http://pdstatic.iem.at/externals-HOWTO>, écrit par Ioannes Zmöltnig.

4. Voir <http://puredata.hurlleur.com>, <http://gmance.org>, <http://codelab.fr/pure-data>, <http://www.puredata.it/forum>, http://www.foroswebgratis.com/foro-pure_data_espana-77988.htm, <http://createdigitalnoise.com/categories/pd-everywhere>.

convention de 2007 à Montréal, *L'œuvre ouverte*, s'est donnée comme ambition, outre le dialogue entre utilisateurs, développeurs et théoriciens, novices et spécialistes, de développer les points suivants :

- mobiliser la création artistique qui emploie les logiciels libres ;
- questionner la diffusion et la distribution libre ;
- cerner les aspects esthétiques et politiques de la culture du logiciel libre ;
- montrer les pratiques ancrées dans une approche multidisciplinaire ;
- mettre en lumière les structures locales de créations à partir du libre ;
- intensifier les échanges entre domaines hétérogènes.

Toutes les conventions (Graz, 2004 ; Montréal, 2007 ; Sao Paulo, 2009 ; Weimar - Berlin, 2011) sont documentées par les écrits des interventions, des photographies, récemment des vidéos. L'IEM a publié les actes de la première convention¹.

Par ailleurs, la communauté s'appuie sur des espaces et des événements où les utilisateurs se forment et se rencontrent.

4.2 Les espaces de formations

Labs et friches culturelles. De nouveaux lieux culturels (hacklabs, friches culturelles, espaces de co-working, associations de logiciel libre) proposent des ateliers *Pure Data*. Ces espaces de création sont très singuliers du fait qu'ils sont issus des acteurs qui les composent, de l'espace géographique auquel ils appartiennent. Ces espaces inventent leurs propres grammaires ancrées dans un contexte socio-politique culturel contemporain. Ils ne copient ni les centres d'animation municipaux, ni les centres d'art contemporains ouverts dans « les années 1980 ». Il n'y a pas véritablement de politique stricte qui les unit. Les notions d'émancipation, d'épanouissement personnel et collectif ne sont pas suffisantes pour les décrire. Certains se distinguent par la volonté d'auto-gestion, d'autonomie, d'autres la promotion de la création émergente, de la sensibilisation des publics à la culture informatique et technique, aux questionnements politiques, éthiques qui leur sont liés. Espaces ouverts, d'échanges, de débats, ils « travaillent » à reconstruire les bases du « vivre-ensemble »

1. Bang, Fribourg, Frank ZIMMER (éd.), 2004.

autour d'une « création émergente » par la prise de parole de chacun. Ils participent à l'essor de l'innovation, accompagnant des démarches associatives, entrepreneuriales d'industrie créative, de développement durable ou de simple développement personnel intégré à une démarche collective. Des espaces mis à disposition où des individus partagent des intérêts communs et développent des projets collectifs. Les formations peuvent être gratuites ou rémunérées. Les formations ou les rencontres sont orchestrées par des artistes ou les membres de ces espaces. Le dialogue autour de questions artistiques, esthétiques, politiques, économiques, sociales, philosophiques en est largement favorisé, dépassant le seul apprentissage technique du logiciel. L'apprentissage est dynamisé par les projets des auditeurs. Les projets sont mutualisés, mis en partage lors d'événements.

Les festivals. Une sensibilisation à *Pure Data* peut se dérouler aussi lors de festivals éponymes. Ces festivals promeuvent et diffusent la création contemporaine, et notamment les créations liées au numérique. C'est l'occasion alors de faire découvrir les outils, les processus utilisés par les artistes. Le logiciel libre employé par les artistes montre les potentialités d'échanges entre artistes, créateurs et grand public et les questions citoyennes qui y sont attachées. La défense des logiciels libres conduit à déplacer des acteurs internationaux pour un événement local. Les spécialistes, les experts se mêlent au grand public. Des générations, des classes socio-professionnelles hétérogènes se rassemblent, dépassant leurs motivations divergentes. Ces festivals contribuent à développer des pratiques amateurs comme professionnelles. C'est aussi un moyen de promouvoir des ateliers sur le long terme, fidéliser des publics, acquérir de nouveaux publics, jauger la popularité de ces pratiques et les capacités à combiner et développer une activité sociale, culturelle, citoyenne et économique autour d'elles.

Les universités. L'apprentissage des logiciels libres dans certaines universités n'est toujours pas la règle en 2012. Ceci est justifié par le fait que le monde du travail emploie majoritairement des logiciels propriétaires. *Pure Data*, pour son usage très spécifique est cependant enseigné dans les universités ou écoles d'art françaises. Les professeurs de *Pure Data* ont des sites qui contiennent le contenu de leurs cours ouverts à la

fois à leurs étudiants et aux internautes. Douglas Edric Stanley, professeur d'Arts numériques à l'école supérieure d'art d'Aix-en-Provence a écrit que son action pédagogique en ligne converge avec l'esprit de l'informatique et d'internet issu de la contre-culture¹ :

Ils sont également le témoignage d'un esprit de partage de l'atelier, d'une promesse de communauté, et d'une revendication : la programmation est une forme de résistance.²

Les projets issus de *Pure Data*. La communauté *Pure Data* est extrêmement diverse et rassemble des projets très hétéroclites. Elle fédère différents types d'utilisateurs ayant des degrés hétérogènes de participation et issus de formations, de professions différentes : musiciens, artistes, scientifiques, créateurs numériques, développeurs soutenus ou appartenant à une institution de formation (professeurs et étudiants) ou entrepreneuriale. En 2004, dans une étude de la communauté *Pure Data* à partir d'une liste de diffusion, Andrea Mayr³ dévoile qu'une partie des contributeurs de *Pure Data* se rémunère de cette activité.

Les Conventions sont le moment de découvrir les créations *Pure Data* les plus évoluées d'un point de vue scientifique, technique, artistique, social, de l'innovation, pédagogique. Les productions les plus lourdes d'un point de vue de la programmation sont des indépendants ou des chercheurs qui enrichissent le logiciel, en développant de nouvelles bibliothèques. Certains développeurs mettent en scène leurs créations. Leurs productions sont souvent mises en accès sur des sites de gestion et d'hébergement de fichiers comme SourceForge, Gitorious, Github. . .

Pure Data est désormais employé dans de nombreux domaines artistiques : théâtre, danse, Vjing, musique, installation, art sonore, performance, art des nouveaux médias, vidéo. . . Le code créatif dépasse largement cette sphère, touchant autant aux sphères scientifiques, d'innovations, que les projets pédagogiques. L'ensemble de ces producteurs échangent des patches, c'est-à-dire des programmes inventés à travers les

1. Dominique CARDON, *La démocratie internet. Promesses et limites*, Paris : Seuil, 2010, pp. 21-26.

2. <http://www.ecole-art-aix.fr/rubrique11.html>.

3. <http://puredata.info/groups/pd-graz/label/bookpdasOpenSourceCommunity>.

plates-formes d'hébergement de programmes, des wikis ou leurs propres sites internet.

Certaines des expériences créatives sont filmées et partagées via le site *Pd-Community Site* ou à partir d'une chaîne du réseau social de vidéos réalisées par les internautes : Vimeo, You Tube ou Daily Motion. Voici quelques exemples de ces créations : les performances des hommes orchestres, Dan Wilcox¹ ou Onyx Ashanti². Le « cowboy robot », Dan Wilcox, joue de la guitare. Ces riffs sont traités à partir d'algorithmes en temps réel liés à une interaction visuelle qui se déploie sur son masque écran. Onyx Ashanti danse et joue de la funk à partir d'un micro-synthétiseur, un saxophone dématérialisé, une sorte de boîte à rythme. Trois capteurs analysent le mouvement, le souffle, les pressions. Ces informations sont reliées à un laser vert-fluo, attachés à son corps. Ces deux artistes donnent accès aux documentations de leur projet artistique et les rendent libres.

Iohannes Zmölnig³ délivre intentionnellement son code dans des performances de *live-coding*. Il s'agit d'une performance basée sur l'improvisation et la programmation informatique où l'écran d'ordinateur du développeur est montré à tous⁴, dévoilant alors le fonctionnement, la logique informatique de ce qui est entendu.

Les projets artistiques réalisés avec *Pure Data* peuvent être engagés politiquement, en donnant par exemple un regard caustique sur les mécanismes de la finance. Les installations du collectif RYBN⁵, les robots traders du projet AntiDataMining, à partir d'algorithmes et de logiciels reconstituent les technologies de transaction en temps réel, en continu et modélisent les fluctuations boursières comme « contrôleurs de hasard ». Elles mettent ainsi le doigt sur les dysfonctionnements de l'économie mondiale.

1. http://www.uni-weimar.de/medien/wiki/PDCON:Concerts/Dan_Wilcox.

2. http://www.uni-weimar.de/medien/wiki/PDCON:Concerts/Onyx_Ashanti et <http://onyx-ashanti.com>.

3. « Do sinusoids dream of electric sweeps », URL : <http://umlaeute.mur.at/Members/zmoelnig/projects/sinusoiddreams>.

4. *TOPLAP Temporaire, transnational, terrestre, transdimensionnel* fut un collectif informel qui souhaitait que les programmes informatiques soient montrés lors de spectacle vivant ou performance, employant des systèmes informatiques. Voir <http://toplap.org/>.

5. <http://www.rybn.org>.

D'un autre côté, ces pratiques artistiques tentent de pousser le logiciel jusqu'à ses retranchements technologiques. Le collectif CHDH réalise des performances synesthésiques se développant à travers des instruments audiovisuels. Deux étapes composent son travail : la création d'instruments, puis leur apprentissage. Les participants créent des algorithmes qu'ils mettent en œuvre dans les matières sonores et audiovisuelles. Leur esthétique se déploie au travers du mouvement, en explorant, développant les potentiels et limites du logiciel *Pure Data*.

La création issue de ce logiciel libre est diverse. Le fait que ce logiciel soit libre donne des voies qui sont singulières à cette création : la documentation, la transparence des programmes informatiques inventés, une volonté de *performer* le logiciel, une forme d'activisme au contexte socio-politique.

Les projets appliqués. Il existe par ailleurs des projets qui télescopent actuellement *Pure Data* dans les nouveaux circuits des nouvelles industries culturelles : logiciels de musique, jeux vidéo, applications pour mobiles. Ces usages permettent de sortir des petits cercles fermés de pratiques pour atteindre le grand public.

Katja Vetter a créé deux logiciels de musique, Instant Decomposer et Slice-Jockey, téléchargeables gratuitement sur son site¹ et présentés avec une documentation ludique sur les concepts scientifiques et techniques interpellés. Le jeu vidéo Spore (action, stratégie et jeu de rôle), dont le thème est le développement des espèces dans l'univers², ou l'application RJDJ (Jockey Reality), jeu d'expérimentations sonores pour I Phone et Android, intègrent un environnement *Pure Data*.

Pure Data se lie encore à des programmes écrits en Java, Objective-C, Python avec la Libpd. Il est cité dans les sites de réalisation de robots amateurs. Il est ainsi employé pour contrôler des directions ou pour saisir les images de la webcam porté par son robot. Dans ce type d'application, *Pure Data* rend la programmation très rapide³.

1. <http://www.katjaas.nl>.

2. Spore a été conçu par le créateur des Sims, et développé à partir d'Open Frameworks. Cf. Kent JOLLY, *Using Pure Data in Spore and Darkspore*, Weimar : Pd-Convention, 2011.

3. <http://letsmakerobots.com/node/6815>.

Pure Data, la recherche et les projets pédagogiques. Bien sûr, *Pure Data* continue à être employé dans la recherche scientifique musicale, ce pour quoi il a été créé. Miller Puckette œuvre toujours au projet initial : la réalisation de partitions se dessinant en temps réel. Il prolonge par cette démarche les intuitions des compositeurs Iannis Xenakis ou Arnold Schönberg sur la question ¹.

Dans la recherche en audiovisuel, *Pure Data* permet de rendre compte de la variété des traitements que l'on peut effectuer sur un signal audiovisuel ².

Le laboratoire de l'université de Sheffield, travaillant sur l'interaction homme-machine à propos de la communication orale, emploie *Pure Data* pour simuler des comportements oraux animaux. *Pure Data* est reconnu pour sa vitesse de calcul et donc la qualité de la simulation du temps réel. Les discours paralinguistiques sont remarquables chez tous les êtres vivants, *Pure Data* permet de les simuler.

Pure Data a été employé pour développer des projets pédagogiques sensibilisant le public à la pratique créative numérique. La Valise Pédagogique Création Interactive ³ ou La MIAM, la mallette interactive artistique multimédia, en sont des exemples. Dans un premier cas, La Valise Pédagogique Création Interactive rassemble du matériel technique, logiciel et documentaire pour la formation à la création au sens large à partir des technologies en temps réel. Elle est modulable en tant qu'outil de création pour de nombreux projets éducatifs, ou en tant que cadre d'apprentissage pour un atelier d'initiation. Dans le second cas, la mallette pédagogique ⁴ MIAM est un ordinateur doté de l'ensemble des périphériques pour la création numérique avec webcams, capteurs, joysticks, wiimotes, cartes Arduino et possédant des ressources numériques pour aborder l'art numérique et son histoire.

1. Conférence de Miller Puckette lors des *Journées du code créatif*, 28-29 mars 2012, Siéréolux, Nantes.

2. Laurent MILLOT, *Traitement du signal audio-visuel. Applications avec Pure Data*, Paris : Dunod, 2008.

3. Créé de 2010-2011 par Jean-Noël Montagné (artiste plasticien), Jérôme Abel (artiste développeur) et les électroniciens africains de ENDA Ecopole en partenariat avec Kër Thiossane (Sénégal) et le CRAS (France).

4. <http://lamiam.fr> – projet réalisé de 2010 à 2011 par les associations Ping et Labomedia en collaboration avec la fabrique du livre.

Pure Data est aussi employé pour sensibiliser à la synthèse sonore des enfants de premier et second cycle, par la confection d'habits interactifs¹.

Pure Data est utilisé de prime abord non en tant que logiciel libre mais en tant que logiciel de programmation graphique ou de programmation musicale. Son accès libre donne la possibilité aux institutions associatives, éducatives et aux auditeurs de se doter aisément de cet outil. Sa qualité de logiciel libre initie indirectement à la mise en commun, à la mutualisation du savoir et questionne la nature des biens informationnels.

5. Les atouts de *Pure Data*

L'atout majeur de *Pure Data* est sa communauté. Elle lui confère un facteur déterminant de durée de vie pour un logiciel ; en permanence celui-ci est complété, renouvelé grâce aux usages. Il permet de multiples expériences temporelles, ce qui lui confère une originalité sans concurrence.

5.1 Les apports d'une communauté

Très tôt au début des années 1990, l'IRCAM a créé un forum, afin de rassembler des compositeurs, chercheurs, enseignants du monde entier autour de leurs productions. Le forum démontrait « le caractère international de l'IRCAM »². En 1994, l'IRCAM prend l'initiative d'une version similaire à MAX/MSP libre, Jmax³, environnement de programmation graphique pour créations multimédias et musicales en temps réel, premier logiciel libre de l'IRCAM (soumis à la licence GPL en 1999). Reposant sur le travail de Miller Puckette, il fut mis en libre accès et développé par l'équipe Systèmes temps réel, sous le contrôle de François Déchelle. L'intérêt de l'IRCAM pour un tel développement était la diffusion indirecte de ses logiciels propriétaires, le rassemblement de bêta-testeurs

1. Christoph TRAPPE, « Making Sound Synthesis Accessible to Children », NIME'12, May 21-23, 2012, University of Michigan, Ann Arbor. URL : http://www.eecs.umich.edu/nime2012/Proceedings/papers/181_Final_Manuscript.pdf.

2. Bernadette DUFRENE, éd., *Centre Pompidou, trente ans d'histoire*, Paris : Centre Pompidou, 2007, p. 543.

3. Voir <http://www.dechelle.net/?p=29>. F. Déchelle écrit : « jMax avait une architecture client/serveur de calcul écrit en C, nommé FTS, et une interface graphique écrite en Java/Swing. Une des caractéristiques de jMax est la portabilité supportée par Irix, Linux, Mac OS X et Windows. »

bénévoles, l'émergence d'idées prolongeant le projet et surtout la portabilité de ce logiciel vers des systèmes d'exploitation libres. Jmax fut le logiciel le plus téléchargé de l'IRCAM, cependant le projet s'acheva en 2002, Jmax rassemblait une communauté de développeurs et utilisateurs inférieure à ses concurrents¹ pour pouvoir se développer.

Autour de *Pure Data* gravite une imposante communauté composée à la fois de développeurs et d'utilisateurs. Les premiers créent de nouveaux « objets » dans une multitude de domaines ou contribuent à améliorer sans cesse le fonctionnement général du logiciel. Les seconds l'utilisent dans de multiples cadres artistiques liés, entre autres, à la performance musicale et audiovisuelle ainsi qu'aux installations interactives. En permanence, ce logiciel libre est ainsi enrichi et adapté à toutes sortes de recherches ou productions.

Le privé a compris à quel degré la contribution entre les utilisateurs et développeurs enrichit les logiciels. Dans le cas de Max 6 sous licence privée IRCAM, dernière version privée similaire à *Pure Data*, la société Cycling 74 vante sa communauté, sa documentation importante, ses tutoriels de formation, ainsi que l'échange avec des modérateurs actifs et des développeurs au service des besoins des utilisateurs, pouvant réagir très efficacement lorsqu'un bug est débusqué par un utilisateur. Une importante communauté d'utilisateurs et de développeurs gravite autour de Max 6 : la collaboration consistant essentiellement à s'échanger des *patches*, donner son avis sur des améliorations techniques du logiciel, signaler des bugs techniques. En outre l'entreprise fait valoir ses créateurs et les promeut. Max/Msp agrandit sa communauté en créant des extensions avec d'autres logiciels. Max coopère actuellement avec Ableton Live, logiciel séquenceur musical pour la composition et l'arrangement, en temps réel, très populaire dans l'électro mondial porté par la scène berlinoise.

5.2 Une temporalité singulière

Miller Puckette et Mark Danks ont décroisé *Pure Data* en re-développant celui-ci avec des atouts multimédias à la fin des années 1990.

1. « on parle alors de Max-like. Mais l'IRCAM, faute d'utilisateurs et donc d'une communauté de développeurs, a arrêté le développement de ce dernier. » Voir <http://fr.wikipedia.org/wiki/Max/MSP#jMax>.

Des contributeurs éclairés ont largement affiné depuis cette technologie. Après avoir observé l'engouement de la communauté *Pure Data* pour la possibilité de programmer en temps réel du graphisme, des animations avec une technologie OpenGL¹, Max s'est pourvu de l'extension Jitter pour pallier à ce manque. Sa qualité de logiciel libre a donné au logiciel d'interaction instrument/machine en temps réel une nouvelle dimension, à partir non d'un programme de recherche, mais d'usages.

Pour Miller Puckette, il existe un hiatus, entre l'outil réalisé par un développeur, une équipe de concepteurs et les personnes qui l'emploient, il en résulte un déséquilibre entre les offres et les demandes. Le « logiciel » comme l'exprime Miller Puckette inscrit la démarche, le cheminement, les conceptions de celui qui le réalise. « Que le développeur le veuille ou non, le logiciel se fait irrémédiablement le reflet de sa mentalité et de sa personnalité. »² *Pure Data*, en tant qu'œuvre ouverte, rend possible des renversements d'usage entre concepteurs et utilisateurs. Ce basculement encourage à dépasser les frontières entre les domaines de savoirs, jusqu'à une indifférenciation des rôles entre utilisateurs et concepteurs.

5.3 Le temps long

Ce logiciel libre n'ayant aucune pression en termes de rentabilité, les versions évoluent de manière homogène, en permettant d'implémenter des patches issus de versions précédentes sur des versions récentes ; au contraire des versions propriétaires qui modifient le plus souvent les interfaces et les modalités suscitant ainsi la commercialisation du nouveau produit, déclare Miller Puckette³. De nombreuses autres possibilités prennent corps avec ce type de logiciel, comme celles pour ces œuvres numériques de se propager et ainsi de ne pas disparaître. Une des conditions de la

1. *Pure Data* peut être un lieu d'expérimentation de l'OpenGL, ce qui le distingue alors pour une activité scientifique pointue.

2. Miller PUCKETTE, « Etude de cas sur les logiciels pour artistes : Max/MSP et sur *Pure Data* », dans : *Art++*, sous la dir. de David-Olivier LARTIGAUD, Paris : Hyx, 2011, p. 182.

3. *Stereolux*, Journées du code créatif 29-30 mars 2012. Il évoque aussi que le libre permet d'éviter la *feature creep*, autrement dit l'ajout de fonctionnalités supplémentaires inutiles avec les finalités premières du logiciel ou provoquant des erreurs informatiques avec le système de base. En tant que logiciel libre, celui-ci peut être entièrement copié et développé selon la volonté du nouveau programmeur. *ibid.*, p. 188.

conservation des fichiers Pure Data est leur utilisation¹ ; ceux-ci peuvent être joués, consultés et remixés². Cette conservation est particulière, elle se rapproche des processus de la mémoire humaine ; elle est liée à l'usage et à la multiplication des points de vue sur une même source. De plus elle offre un espace d'expressions des points de vue de chacun. On peut voir l'exemple illustratif de la fondation Daniel Langlois qui se focalise sur la documentation et la conservation du patrimoine des arts médiatiques en mettant en place un site participatif ciblant les artistes, mais également les témoins des pratiques. La clé de la mémoire de ces œuvres est d'« instaurer un dialogue à plusieurs voix entre les différents acteurs concernés³ ». C'est pourquoi le logiciel libre, celui qui donne une ouverture totale et complète à ses sources, avec l'aide de forums et d'espaces présentiels à sa pratique, est la condition nécessaire et maximale à l'expérience du temps long pour des œuvres numériques.

Conclusion

Pure Data repose sur de nombreux concepts et conventions mathématiques et informatiques ; il nécessite un niveau non négligeable de technicité dès qu'un projet, même le plus anodin, se déploie. Toutefois, par l'utilisation de ce logiciel, par sa communauté ou par ses productions créatives, les profanes sont entraînés dans un territoire scientifique, technologique, épistémologique ou artistique et esthétique pour lequel ils éprouvaient de l'indifférence, de l'ignorance, du mépris ou des complexes. Sa programmation, via une interface graphique, rend même l'apprentissage ludique, attrayant et sa simplicité d'usage accélère le processus créatif.

Pure Data est autant une sensibilisation aux sciences, aux technologies, qu'aux travaux de recherche. Les pièces de Philippe Manoury à la fin des années 1980 avec *Patcher* jusqu'aux travaux actuels de CHDH à partir du

1. Ces utilisations sont des signes de démocratie pour Jacques Derrida. « Nul pouvoir politique sans contrôle de l'archive, sinon de la mémoire. La démocratisation effective se mesure toujours à ce critère essentiel : la participation et l'accès à l'archive, à sa constitution et à son interprétation ». Jacques DERRIDA, *Le mal d'archive*, Paris : Éditions de Minuit, 1995, p. 15.

2. doddés, forkés. . . .

3. Anne LA FORÊT, *Le net art au musée*, Paris : Vrin, 2011, p. 161.

GEM sont de cet ordre. Ces travaux sont l'alchimie de deux territoires, art et science, sans qu'il y ait instrumentalisation de l'un par l'autre.

L'apprentissage de ce logiciel est l'expérience d'un processus d'entraide, de solidarité issu des forums, de l'expérience de travail en commun sans organisation hiérarchique. *Pure Data* est aussi un moyen de réunir des personnes aux parcours, aux représentations, aux valeurs, aux objectifs hétéroclites. Son initiation conduit inévitablement à questionner l'environnement numérique contemporain et passé, à éveiller à des problématiques économiques, politiques, sociales, éthiques le concernant. L'apprentissage d'un média est l'occasion de donner un langage, une grammaire conférant une forme, une articulation à une parole spécifique. Elle est la voie de l'émancipation, et de la possibilité d'expression, de délibération de construction du commun, de conversations dépassant un territoire national ou de langues.

Cependant le degré technique à acquérir ne finit-il pas par enfermer les joueurs de *Pure Data* dans la répétition de figures imposées, sans qu'ils puissent s'en affranchir ? Ou au contraire donne-t-il la possibilité aux auteurs de propager, disséminer le contre-pouvoir de la participation directe vers des interlocuteurs ? On observe à quel point ce logiciel est lié aux institutions publiques, à la passion de ses salariés, à ses structures associatives qui créent du bien commun, défendent l'intérêt général. Cette programmation qui donne lieu à de multiples applications Low Tech, conduit à un jeu ludique entre les universités du monde autour de l'interaction instrument/machine du temps réel, des technologies Open GL, ses applications sur les écrans mobiles, son imbrication avec d'autres langages de programmation. D'ailleurs l'interopérabilité est l'une des préoccupations majeures du logiciel libre. La librairie libpd permettant son interopérabilité avec de nombreux langages de programmation informatique montre que *Pure Data* conscientise sa qualité de logiciel libre.

Apprendre un langage, une logique propre à *Pure Data* ne détourne-t-il pas de même les citoyens des langages informatiques usuels et ne les absorbe-t-il pas dans une zone réduite de l'informatique, l'informatique artistique ? La réponse à cette question trouve sa clé dans les objectifs pédagogiques des passeurs. L'artiste Katja Vetter documente par exemple ses travaux en rappelant les concepts mathématiques et physiques présents dans *Pure Data*, à l'intérieur de ces propres *patches*, le cheminement

intellectuel suivi pour créer ces *patches* jusqu'à la création de ses logiciels, outils pour des œuvres sonores et multimédias. Katja Vetter appartient à ces artistes qui créent leurs outils pour produire leurs créations, qui documentent finement leurs processus techniques, qui partagent leurs outils. Étrangement, elle n'exprime pas son parcours esthétique et artistique qui la conduit à cette démarche. La diffusion du logiciel *Pure Data* montre que les outils du libre sont le support de nombreuses dynamiques : dynamique de recherche, de projets pédagogiques, de créations numériques, des industries créatives.

Il est très difficile de faire une prospection sur le devenir de ce logiciel, de son évolution. Le temps réel est pour l'instant à un point très avancé. Le temps réel est désormais travaillé jusqu'à un point où la perception humaine ne le remarque plus. Toutefois dans le cas de *Pure Data*, il peut être affiné pour des usages comme le jeu vidéo ou la captation de la voix humaine.

Le logiciel, même s'il ne s'organise pas sur un principe démocratique à ce jour est accompagné de nouveaux lieux de la culture qui ont des objectifs sociaux, politiques et non plus seulement une prétention esthétique sans instances de critiques. Culture active, actrice, lieu d'apprentissage, des lieux névralgiques de rencontres, de dynamiques économiques, de circulation, de courants de pensée contemporains, de transmission, d'expérimentation de la démocratie directe, peu nombreuses en France contrairement à d'autres pays européens. Les pouvoirs publics n'ont pas encore réalisé et massifié l'élan que ces lieux pouvaient offrir en terme de citoyenneté, de responsabilité, de transformation des consciences. Le logiciel libre *Pure Data* est un modèle d'organisation et de gestion des biens informationnels. Cependant si les modalités d'appropriation ne sont pas conscientisées, si le fonctionnement des structures qui proposent des rencontres ou ateliers *Pure Data* ne sont pas proches des processus organisationnels inspirés du logiciel libre au sens général du terme, les contributeurs ne seraient-ils pas instrumentalisés, en les positionnant comme consommateurs dans un contexte « do it yourself », pour conduire *in fine* à l'altération des coopérations, des mutualisations, de l'entraide, de l'élan créatif autour des technologies ?

Je remercie l'ensemble des acteurs de la formation Pure Data du CRAS proposée par *Mains d'oeuvres* (Jean-Marie Boyer, Cédric Buron, Christian Delecluse, Agnès Le Foulgoc, Jean-Yves Gratius, Joseph Jaouen et tout particulièrement Pascale Gustin) ainsi que toute l'équipe de Framasoft.

Références

- CARDON, Dominique, *La démocratie internet. Promesses et limites*, Paris : Seuil, 2010.
- COMBES, Malika, *L'IRCAM fête ses 30 ans : retour sur une institution culturelle atypique dans ses rapports avec ses organismes de tutelle*, <http://musique.ehess.fr/docannexe.php?id=238> : EHESS, 2007.
- DERRIDA, Jacques, *Le mal d'archive*, Paris : Editions de Minuit, 1995.
- DUFRENE, Bernadette, éd., *Centre Pompidou, trente ans d'histoire*, Paris : Centre Pompidou, 2007.
- GRYNSZPAN, Emmanuel, « Confluence et divergences. La techno face aux musiques savantes », dans : *Musiques, arts et technologies. Pour une approche critique*, sous la dir. de Roberto BARBANTI et al., Paris : l'Harmattan, 2004.
- IRCAM, « Philippe Manoury », dans : *Les Cahiers de l'Ircam* 8 (1995).
- LA FORÊT, Anne, *Le net art au musée*, Paris : Vrin, 2011.
- LANGER, Daniela, *Va-et-vient. Entretiens avec Philippe Manoury*, Paris : Musica Falsa, 2001.
- MAYR, Andrea, « Pd as Open Source Community », dans : *Bang, Pure Data, Berlin, Wolke Verlagsges* (2004), p. 33–44.
- MILLOT, Laurent, *Traitement du signal audio-visuel. Applications avec Pure Data*, Paris : Dunod, 2008.
- MOREAU, Antoine, « Le copyleft appliqué à la création. Une reformulation des données culturelles ? », thèse de doct., Nice : Université de Nice Sophia Antipolis, 2001.
- PACHET, François et Jean-Pierre BRIOT, éd., *Informatique musicale : Du signal au signal musical*, Paris : Hermès, 2004.
- PUCKETTE, Miller, « Etude de cas sur les logiciels pour artistes : Max/MSP et sur Pure Data », dans : *Art++*, sous la dir. de David-Olivier LARTIGAUD, Paris : Hyx, 2011.

Wikipédia et le monde du Libre

Adrienne ALIX

Au sein du monde « libre » figure depuis 2001 un projet singulier, à la fois symbole des principes du Libre et pourtant en grande partie porté par des contributeurs qui ne se sentent pas nécessairement « libristes » : Wikipédia, l'encyclopédie libre et collaborative, qui offre en 2012 près de 20 millions d'articles en plus de 280 langues, tous disponibles sous licence libre Creative Commons BY-SA.

Wikipédia est la fille du Libre, imaginée, pressentie parfois (par R. Stallmann, dans « The Free Universal Encyclopedia and Learning Resource »¹ à la fin des années 1990), lorsque le Libre commence à réellement s'organiser et à sortir du « tout logiciel », c'est-à-dire d'une utilisation des licences libres uniquement dans le cadre de la diffusion et de la réutilisation des logiciels informatiques. Brouillon ouvert et libre d'un projet beaucoup plus académique appelé Nupedia, qui devait être une encyclopédie classiquement écrite par des experts et publiée sur le Web, Wikipédia a très rapidement démontré l'intérêt très concret de l'utilisation des licences libres et du fonctionnement collaboratif pour la création et la diffusion de contenu encyclopédique.

L'usage de Wikipédia et la contribution au projet se sont d'abord répandus dans des milieux spécifiques : informatique, sciences dures, etc. S'y

1. Voir : <http://www.gnu.org/encyclopedia/free-encyclopedia.fr.html>.

trouvaient des utilisateurs de logiciels libres qui, sans doute les premiers, ont compris l'intérêt et la philosophie de Wikipédia : écrire collaborativement et partager librement l'ensemble des connaissances humaines, dans un esprit non-marchand et humaniste. Une large interview menée en 2011 auprès des premiers contributeurs francophones, à l'occasion des 10 ans de Wikipédia, montre l'importance de la connaissance des logiciels libres dans l'accès à Wikipédia entre 2001 et 2003, par les valeurs qui sont communes, mais aussi par la sociabilité du Libre : de nombreux contributeurs indiquent avoir découvert Wikipédia par le biais de forums ou de listes de diffusion rattachés au logiciel libre, ou à la culture informatique de manière plus générale.

Dix ans après, qu'en est-il ? La Wikipédia francophone compte environ 6 000 contributeurs actifs par mois et plusieurs dizaines de milliers de contributeurs occasionnels. Une partie d'entre eux a des convictions qu'on pourrait qualifier de « libristes » : importance du travail collaboratif, de l'ouverture des contenus et de la libre diffusion des connaissances, ainsi que de la maîtrise du processus de création, impliquant une visibilité des sources. Les termes de la licence utilisée sur Wikipédia restent cependant relativement flous pour une part importante des contributeurs. Les effets induits par l'utilisation des licences libres sont acceptés mais l'adhésion aux valeurs du Libre arrive souvent en un deuxième temps et avec une profondeur variable selon les personnes.

Chez les lecteurs de Wikipédia (19 millions de visiteurs uniques par mois en France), cette compréhension du Libre est encore plus légère, la qualification de « Libre » étant davantage perçue comme ce qui permet une libre lecture et une libre contribution, que par ses aspects de réutilisation et modification.

La communauté libriste n'est pas absente de Wikipédia : active, respectée, souvent proche des associations locales (comme Wikimédia France), elle prend une part importante dans la pédagogie et l'explication des licences, tant auprès de la communauté que vers l'extérieur. Elle est là pour rappeler les fondamentaux du projet, servir de référence, amener certains à mieux comprendre le Libre, mais ne peut imposer aux contributeurs une adhésion totale aux valeurs du Libre.

Wikipédia est donc l'expérience d'un noyau initial très proche du Libre, mais qui s'est adjoint des forces contributives ignorantes de ses valeurs,

qui collaborent au projet pour des raisons extrêmement variées, mais pas majoritairement pour des raisons militantes.

À travers l'ensemble d'interviews réalisées en 2011, mais aussi en tirant des enseignements de l'observation de la communauté et de ses réactions face aux principes et aux contraintes des licences libres depuis les débuts de Wikipédia, nous essaierons de montrer l'évolution de la place du Libre sur Wikipédia et des rapports de la communauté wikipédienne avec le Libre.

1. De Nupedia et GNUpedia à Wikipédia, un projet né sur les bases du logiciel libre

Le projet Wikipédia est né un peu par hasard, à partir des expérimentations de Jimmy Wales et Larry Sanger, alors engagés dans la création d'une encyclopédie numérique traditionnelle, Nupedia.

Jimmy Wales est en 2000 un entrepreneur du Web, fondateur du moteur de recherche érotique Bomis et ancien analyste financier, qui décide d'investir ses bénéfices dans la réalisation d'un rêve d'enfant : une encyclopédie. Familier d'Internet et conscient des potentialités du Web qui commence, à la fin des années 1990, à devenir réellement populaire, il envisage son projet comme une encyclopédie numérique, traditionnelle dans son contenu mais diffusée librement sur le Web. S'adjoignant les services d'un étudiant en philosophie, Larry Sanger, ils créent en septembre 2000 le projet d'encyclopédie Nupedia.

Nupedia fonctionnait au départ sur le logiciel libre NupeCode, spécialisé dans les projets de peer-review (examen par les pairs) de masse mais souffrant dès le départ de manque de fonctionnalités et d'agilité¹. Un projet bien éloigné de ce que sera Wikipédia, puisque la notion d'auteur d'un article est fondamentale et repose sur des qualifications pré-requises, ensuite soumises à une évaluation par des pairs, dans la tradition de l'édition scientifique, au contraire de Wikipédia où la révision des textes est faite par les lecteurs-contributeurs, où la notion d'auteur se fait collective

1. Voir <http://en.wikipedia.org/wiki/NupeCode>.

et où la qualité de l'auteur se mesure à ce qu'il écrit et non à ce qu'il est. Le projet est placé sous une licence libre créée pour lui, la *Nupedia Open Content License*. Richard M. Stallmann et la Free Software Foundation, très intéressés par le projet, convainquent début 2001 Jimmy Wales et Larry Sanger de placer le contenu de Wikipédia sous licence GNU-FDL (GNU Free Documentation License).

Dès le démarrage de Nupedia, et comme le confirme Larry Sanger dans son mémoire sur l'histoire de Nupedia et Wikipédia, il était clair que le projet devait être ouvert et calqué sur le principe des logiciels libres afin de pouvoir être largement approprié par les rédacteurs volontaires et qu'il bénéficie des plus larges possibilités de diffusion¹.

Mais le processus d'édition et de validation sur Nupedia souffre de lourdeurs, d'une bureaucratie qui semble aux participants excessive et surtout peu cohérente avec les principes du logiciel libre. Richard M. Stallmann, entre temps, crée son propre projet, GNUPedia, en janvier 2001 (la confusion sera souvent faite entre les deux projets à l'époque, mécontentant notamment Jimmy Wales). Les participants des deux projets semblent se recouper et attestent de la lenteur du développement de Nupedia, qui ne compte qu'une vingtaine d'articles terminés un an après sa création.

C'est alors qu'intervient un paramètre technique important : le logiciel wiki.

Le « wiki » est un type de logiciel développé en 1995 par Ward Cunningham, un informaticien américain souhaitant favoriser le travail collaboratif, l'intervention de l'internaute lors de sa lecture d'un site Web, et transposer dans les contenus Web les techniques du développement logiciel. Révolutionnaire à l'époque, il permet une écriture collaborative, un historique de chaque contribution, une publication immédiate sur Internet et fonctionne avec une interface simple (pour l'époque) puisque n'utilisant que des balises très basiques. Dans un contexte où le WISIWYG n'existe pas encore réellement pour la mise en ligne de contenu, le wiki permet

1. « Now, Jimmy was quite clear that he wanted the project to be in principle open to everyone to develop, just as open source software is (to an extent) ». Larry SANGER, « The Early History of Nupedia and Wikipedia : A Memoir », dans : *Open Sources 2.0 : The Continuing Evolution*, sous la dir. de Chris DiBONA, Mark STONE et Danese COOPER, Sebastopol : O'Reilly Media, 2005, p. 307-338.

(NdE : version du 18/04/2005, disponible sur <http://features.slashdot.org/story/05/04/18/164213/the-early-history-of-nupedia-and-wikipedia-a-memoir>.)

à des internautes non formés à l'écriture HTML de contribuer facilement sur un site.

Jimmy Wales et Larry Sanger tentent donc une expérimentation visant à ouvrir, sur un wiki, la possibilité à tout internaute de démarrer un article encyclopédique, l'idée étant d'amorcer le travail pour permettre ensuite aux experts de le compléter et de le placer sur Nupedia. Au départ, il n'est pas question de faire passer les articles écrits sur le wiki pour de « vrais » articles encyclopédiques, on est dans l'expérimentation et la découverte. Sur la liste de diffusion [Nupedia-l], dans un message intitulé « Let's make a wiki », Larry Sanger présente le système du wiki le 10 janvier 2001 comme quelque chose d'expérimental :

Sur la page d'accueil du wiki de Nupedia nous devons être ABSOLUMENT clairs sur le fait que c'est expérimental, que les contributeurs de Nupedia n'ont pas de contrôle sur ce qui se passe, et que la qualité des articles, la discussion, etc., ne doivent pas être pris comme un reflet de la qualité des articles, des révisions, etc. sur la partie principale du site Nupedia.

Ce wiki expérimental devait être accessible depuis Nupedia sans pour autant en faire partie, tout en reconnaissant que c'est un pas supplémentaire vers l'ouverture du projet, l'arrivée de nouveaux contributeurs et une nouvelle manière de travailler ensemble :

En ce qui concerne l'utilisation d'un wiki sur Nupedia, c'est l'« ouverture » ULTIME et un format simple pour développer du contenu... qui peut être un endroit où des changements additionnels et des commentaires peuvent être recueillis [...] Cela peut donner des choses très intéressantes pour beaucoup d'utilisateurs, qui peuvent *instantanément* créer, éditer et commenter des articles.

D'emblée, il est décidé que les contributions faites sur ce wiki seraient placées sous licence libre.

Le wiki, en tant que logiciel, n'est à l'origine choisi par Wales et Sanger que comme un outil, une manière de dynamiser la production d'articles encyclopédiques. Toute la culture qui va avec le wiki n'était pas prise au sérieux. Ward Cunningham, l'inventeur du Wiki, avait pourtant prédit à Wales et Sanger que Wikipédia porterait très fortement l'empreinte de la

culture wiki, au point que ses deux caractéristiques principales (être une *encyclopédie*, et être *produite sur un wiki*) seraient inextricables. Et en effet, Larry Sanger peut détailler quelles sont les influences principales du système wiki sur Wikipédia :

D'un côté la culture wiki est typique, le logiciel wiki encourage effectivement l'ouverture extrême et la décentralisation : l'ouverture puisque (c'est la façon dont le logiciel est conçu) les modifications de pages sont enregistrées et visibles publiquement, et (en général) que les pages peuvent être modifiées par n'importe qui ; la décentralisation, parce qu'il n'est pas nécessaire pour une personne ou un organisme d'organiser le travail à faire, on pourrait plutôt dire que le travail se fait de la manière dont les gens le souhaitent, et quand ils le souhaitent. Le logiciel wiki décourage aussi (ou du moins ne facilite pas) l'exercice de l'autorité, car l'avancement des travaux se fait sur n'importe quelle page, et sur n'importe quel grand wiki actif, il y aurait trop de travail à faire pour un superviseur ou un groupe restreint de superviseurs pour garder la main. Ces caractéristiques sont toutes devenues des caractéristiques de Wikipédia.¹

Cette description se rapproche de la culture libriste par toute l'ouverture qu'elle décrit, la prise en main par les internautes de la gestion collective du contenu, la contribution volontaire et ouverte à tous, le suivi des modifications et l'accès aux sources des articles, mais diffère principalement par le fait que dès l'origine les potentialités du wiki interdisent la « gestion de projet » et la direction du travail. Il n'existe pas non plus de « commit », de passage d'un contenu « proposé » à un contenu validé et implémenté au cœur du système : d'emblée, la contribution est visible à tous, intégrée dans le contenu général du wiki.

Wikipédia naît donc officiellement, en tant qu'entité associée mais non confondue de Nupedia, le 17 janvier 2001. Larry Sanger invite les participants à Nupedia (environ 2 000 à cette date) à tester l'outil et à écrire un petit article :

Ça ne vous prendra que cinq ou dix minutes ». Le site est accessible à l'adresse <http://www.wikipedia.com/>.

1. *ibid.*

Très vite, Wikipédia prend un dynamisme que Nupedia n'avait jamais eu, la nouvelle d'un projet original se répand au sein des communautés scientifiques anglo-saxonnes, mais aussi au-delà, dans des milieux scientifiques et informatiques en lien avec le monde anglo-saxon. L'idée de versions linguistiques autonomes est très vite adoptée et une version germanophone de Wikipédia ouvre le 16 mars 2001, suivie par le catalan quelques heures après, et le français le 23 mars. Les autres langues suivent à un rythme régulier, jusqu'à avoir près de 285 versions linguistiques autonomes de Wikipédia fin 2012 ¹.

Le projet Nupedia perd peu à peu sa raison d'être et ses contributeurs préfèrent contribuer uniquement sur Wikipédia. Le 1^{er} mars 2002, Larry Sanger démissionne de son poste de rédacteur en chef de Nupedia. Le projet Nupedia lui-même meurt en septembre 2003, après un arrêt inopiné des serveurs qui ne seront pas relancés.

Simultanément, Wikipédia fait se rejoindre plusieurs projets et utopies jusqu'alors non reliés (même si déjà réunies en pensée) : le projet d'encyclopédie libre de Richard Stallman déjà évoqué, mais également le projet Xanadu imaginé dans les années 1960 par Ted Nelson et voué à un partage instantané et universel des données ², lui-même inspiré par le projet de classification universelle des connaissances et le répertoire bibliographique universel pensés et mis en œuvre (à l'époque sur papier) par le belge Paul Otlet à la fin du XIX^e siècle ³.

2. Les Wikipédiens, une communauté issue du monde du Libre

Le projet Wikipédia, comme on l'a vu, n'est pas né des mains de créateurs eux-mêmes engagés dans la communauté du logiciel libre. Ni Jimmy Wales, ni Larry Sanger ne sont des membres investis de la communauté

1. Voir la liste des versions linguistiques de Wikipédia à jour sur http://meta.wikimedia.org/wiki/List_of_Wikipedias.

2. Voir sur Wikipédia l'article concernant le Projet Xanadu : http://fr.wikipedia.org/wiki/Projet_Xanadu.

3. Voir sur Wikipédia les articles sur Paul Otlet (http://fr.wikipedia.org/wiki/Paul_Otlet) et le Répertoire bibliographique universel (http://fr.wikipedia.org/wiki/Répertoire_bibliographique_universel).

libriste américaine : ils ne sont pas développeurs, ne sont pas connus pour un engagement particulier en faveur du logiciel libre ou des libertés sur Internet. L'homme d'affaire et le doctorant en philosophie n'appartiennent pas à un groupe formalisé ni ne se revendiquent de ces groupes même s'ils semblent en connaître les figures marquantes comme Richard Stallman. En ce début des années 2000, la communauté libriste n'est de toute façon pas organisée pour d'autres actions que le développement logiciel et communique de façon informelle, essentiellement par le biais de forums et de listes de discussion. Internet n'a pas encore pris la place prépondérante qu'il a aujourd'hui auprès du grand public, et hors du monde de l'informatique, la question des logiciels et des licences libres n'a pas de visibilité. Aux États-Unis, si la Free Software Foundation existe depuis 1985 et l'Electronic Frontier Foundation depuis 1995, leur influence n'est, à l'époque, pas réellement sortie hors d'un milieu universitaire et technologique très particulier. Cela étant, les principes du logiciel libre sont déjà posés et définis de façon très claire en 2000-2001, et cela permet à Wales et Sanger d'avoir avec Richard Stallman des rapports de compréhension et de coopération, notamment sur le domaine de la licence applicable à Nupedia puis Wikipédia.

Se pose, avec la naissance de Wikipédia, la question d'une communauté : contrairement au projet Nupedia où le critère d'entrée est la spécialité, l'expertise d'un contributeur, pour Wikipédia il est au moins aussi important d'avoir envie de participer à un projet un peu utopique, sortant des sentiers battus, et très nettement proche du mode de création et de production d'un logiciel libre : travail collaboratif, revue permanente par la communauté, partage des connaissances, organisation horizontale, bénévolat et investissement personnel.

C'est donc assez naturellement que le projet se diffusa d'abord au sein de la communauté des développeurs et informaticiens déjà familiers avec les valeurs des logiciels libres, et partisans des licences libres.

En 2011, à l'occasion des 10 ans de Wikipédia, une interview de quelques contributeurs parmi les pionniers de Wikipédia a été réalisée sur la Wikipédia francophone ¹.

1. Tout d'abord synthétisée sous la forme d'un billet de blog publié sur le blog de Wikimedia France à l'occasion des 10 ans de Wikipédia (<http://blog.wikimedia.fr/wikipedia-a-10-ans-2685>), l'intégralité des réponses a été ensuite publiée directement

Destinée à faire revivre, sans prétention statistique, le profil des premiers wikipédiens, elle a livré un certain nombre d'informations extrêmement éclairantes sur l'esprit des premiers contributeurs et leur origine, situant notamment celle-ci au cœur des problématiques des communautés libristes de l'époque.

Le choix des personnes interrogées s'est porté sur des contributeurs ayant rejoint Wikipédia entre 2001 et 2003, à une période où le projet était encore balbutiant. La première question qui leur était posée était de savoir par quel biais ils avaient eu connaissance de l'existence de Wikipédia. Les réponses sont quasi unanimes :

Rinaldum — Je me souviens assez bien de tout ça. Un jour en me baladant sur le site de la FSF (horriblement moche à l'époque) je suis tombé sur le projet Gnupedia si je me souviens bien, qui devint Nupedia, mais alors que je suivais avec intérêt son développement j'eus le cœur serré quand je me suis rendu compte que le projet périssait doucement... jusqu'au jour où j'ai suivi un lien vers une version différente de Nupédia où tout le monde pouvait participer ! Je suis arrivé en fait sur www.wikipedia.com géré par une société privée américaine... c'était très prometteur, et ça correspondait à ce que je voulais faire !

Bobby — Je ne me rappelle plus comment ni quand je suis tombé sur ce site, sans doute au hasard de mes recherches. Wikipédia faisait déjà sensation dans le milieu « Libre », projet titanesque qui faisait des petits dans le monde entier.

sur Wikipédia à titre d'archive communautaire (http://fr.wikipedia.org/wiki/Wikipédia:10e_anniversaire_de_Wikipédia/interview).

Youssefsan — J'avais découvert Nupédia et je pense aussi Wikipédia via une revue consacré à l'informatique ou Linux. J'avais décidé de participer à Nupédia. Mais en réalité, je n'y ai pas fait grand-chose si ce n'est échanger quelques courriels. Puis, j'ai vu l'annonce de Rinaldum sur DLFP (Da Linux French Page) (<http://linuxfr.org/2002/02/12/7070.html>) et je suis venu. Je m'étais aussi initié au système « wiki » par le biais du wiki qu'hébergeais alors DLFP.

Yvesb — J'ai découvert Wikipédia en mars 2003. J'avais commencé depuis plus d'un an à participer à des projets *open source*. Linux commençait à sortir de l'anonymat. On sentait comme un vent de fraîcheur traverser sur le monde du logiciel dans lequel je nage.

Polletfa — C'était en 2001 et à vrai dire, non, je n'en ai pas un souvenir très précis. Wikipédia n'était à l'époque pas très connue du « grand public » mais on en parlait beaucoup dans le milieu du logiciel libre, et je passais à l'époque beaucoup de temps sur des sites ou forums consacrés au libre. [...] C'est par le logiciel libre que j'ai découvert Wikipédia, et en 2001, c'était le cas de beaucoup de contributeurs je pense. C'est l'idée de transposer les concepts du logiciel libre à d'autres domaines, en dehors de l'informatique, que je trouvais séduisante.

Lionel Allorge — Étant impliqué dans le logiciel libre, j'ai entendu parler de ce projet d'encyclopédie basée sur la licence de do-

documentation libre. Il est probable que ce soit via des échanges sur des listes de discussion. [...] Le projet me semblait la suite logique des logiciels libres qui encouragent le libre partage du savoir.

Kelson — J'ai entendu parler de Wikipédia sur linuxfr. [...] Venant du logiciel libre et du monde de l'informatique, j'étais idéologiquement et pratiquement prédisposé à aimer Wikipedia. Trouver Wikipédia formidable allait donc de soi pour moi.

La place du forum DLFP (ancien nom de Linuxfr.org) semble essentielle comme porte d'entrée de Wikipédia pour les francophones. Plusieurs contributeurs ont indiqué être venus suite à la lecture de dépêches sur ce forum. Le 12 février 2002, l'annonce de l'existence de la version francophone de Wikipédia est ainsi faite, anonymement et de façon lapidaire ¹ :

Anonyme — C'est juste pour vous signaler que Wikipedia est maintenant en français et qu'ils ont besoin de beaucoup de monde. Wikipedia est un projet d'encyclopédie libre et écrite coopérativement . La section linux est à bâtir alors à vos claviers. Le site anglais a déjà plus de 23 000 textes et le français seulement 100. Tout est en GPL même le moteur. [Note du modérateur : Wikipedia est une encyclopédie qui complète sa grande sœur Nupedia. Les deux projets sont soutenus par la FSF].

Cela étant, la place des forums « de geeks » (informatique non spécifiquement libre, jeux, etc.) est souvent soulignée aussi, ce qui a d'emblée fait entrer dans le noyau initial de la communauté wikipédienne des non-libristes. Ainsi, Florence Devouard, qui a pris une place très importante

1. <http://linuxfr.org/news/wikipedia-a-besoin-de-toi>.

dans la communauté jusqu'à être présidente de la Wikimedia Foundation entre 2006 et 2008, n'a pas découvert Wikipédia par le biais du libre :

Florence Devouard — À l'époque, je ne faisais pas du tout partie du mouvement libre. En fait, je n'avais jamais entendu parler de licence libre et j'aurais bien été incapable de définir « droits d'auteur ».

Anthere — Wikipédia était complètement inconnue, à l'exception des libristes peut-être. Autant dire qu'il n'y avait pratiquement aucune chance de tomber dessus par hasard. [...] Fan de jeux [...] je me connectais parfois à des forums pour discuter avec d'autres joueurs. À cette occasion, été 2001, j'ai rencontré un Canadien anglophone, qui était un des premiers participants de Wikipedia (en anglais donc). [...] Le canadien et moi avons commencé à discuter jeu, stratégie, puis dérivés sur le concept de Gaïa, sur les OGM, la biosécurité... et finalement, il m'a traînée sur Wikipedia. [...] Finalement, mon premier article sur la Wikipedia en anglais doit dater de janvier ou février 2002. J'ai vite découvert la version francophone. Il n'y avait personne dessus. Une dizaine d'articles, essentiellement pour nous guider.

Cette arrivée sur Wikipédia par les forums « geeks » est relatée par d'autres contributeurs :

Hashar — En parcourant un des nombreux forum d'Usenet, je suis tombé sur un message d'un utilisateur dont la signature comportait un lien vers wikipedia.com. Un clic et me voilà happé par le wiki.

Poulpy — Le nom du projet est revenu à plusieurs reprises sur les forums que je fréquentais (des trucs de geek, forcément) et j'ai fini par faire quelques éditions sous IP à la fin de l'année. Je me suis créé un compte par la suite.

Ryo — J'ai découvert Wikipédia à travers une notice sur le site Slashdot, notice qui disait sans doute quelque chose à propos du besoin de bénévoles pour écrire des articles, ou alors un cap d'articles qui avait été atteint, je ne sais plus. Du coup je suis allé voir, et j'ai immédiatement accroché. Je me suis inscrit, et j'ai commencé à contribuer.

Lorsque la question de la communauté de valeurs entre Wikipédia et le monde du logiciel libre est abordée, les réponses sont tout aussi claires : Wikipédia est une déclinaison, une extension des valeurs du logiciel libre, où prend place, de façon importante, la notion de partage du savoir, des connaissances, peut-être de façon plus marquée que dans le monde du logiciel : là où on ouvre son code et on laisse ensuite ceux qui le voudraient venir le consulter, le réutiliser, l'améliorer, Wikipédia apporte une notion d'organisation de la diffusion : tout le monde partage au même endroit, met en place une organisation des articles, et fait ouvertement la promotion de cette notion de diffusion. Le lecteur peut se contenter d'être lecteur et de demeurer consommateur de contenu, de façon beaucoup plus claire que dans le monde du logiciel. Le seuil de compétences requis pour participer à Wikipédia, en tant que lecteur ou en contributeur, s'abaisse considérablement. C'est une révolution intellectuelle pour beaucoup, ce que certains attendaient, ou espéraient, ou n'auraient pas osé imaginer :

Anthere — Il s’agissait plutôt de partage de connaissances et à l’époque, beaucoup de partage de l’activité rédactionnelle.

Kelson — Je me suis investi plus car j’y ai vu rapidement la promesse de voir les principes du logiciel libre dépasser le cadre de l’informatique. C’est donc pour éviter que (je cite) « Ce serait peut-être l’une des plus grandes opportunités manquées de notre époque si le logiciel libre ne libérait rien d’autre que du code » que je me suis investi vraiment ; il fallait que « ça marche ». La motivation était donc avant tout politique ; j’y ai aussi pris beaucoup de plaisir.

Hashar — C’est lié à ma personnalité et aux fondements d’Internet. Conçu par l’armée américaine, les universitaires se le sont vite approprié pour diffuser leurs travaux de recherche. Le mouvement du logiciel libre partage lui aussi des connaissances, il manquait un outil pour le savoir d’une manière générale.

Aoineko — C’était un lieu où nous pourrions travailler ensemble pour que la somme de nos connaissances aboutisse à des articles plus riches, plus exhaustifs et plus neutres que tout ce qu’on pouvait trouver ailleurs. D’où la théorie du $2 > 1 + 1$ à laquelle je crois profondément. Pour beaucoup, cela semblait très utopique, mais j’avais l’intime conviction que cela était possible (même si nous ne pouvions évidemment pas imaginer l’ampleur qu’a pris le projet aujourd’hui !). [...] Par contre à mes débuts, la notion de Libre était assez floue pour moi et ne constituait pas une motivation particulière. Ce n’est qu’en participant à Wikipédia que j’ai pris conscience que certaines dérives

du système de droit d'auteur pouvaient aller à l'encontre de l'intérêt commun ; notamment, celui de la libre diffusion de la connaissance.

Bobby — Le partage des connaissances fait partie des motivations et ne revêt aucun aspect politique de diffusion de la culture ou des logiciels libres mais seulement pratique. Le fait même que wikipédia fonctionne et se développe prouve que les logiciels libres sont crédibles, la dimension de wikipédia appuie le fait que les logiciels libres tiennent une place importante et ne peuvent plus être ignorés. La libre diffusion de la culture n'est qu'une conséquence du succès de Wikipédia. Il faut donc y participer.

Yvesb — La seule encyclopédie digitale largement diffusée était une petite poussive Encarta de Microsoft. La promesse de Wikipédia c'était l'accès au plus grand nombre. Je n'imaginais pas que ça puisse exploser comme maintenant.

Wikipédia semble donc être apparue aux yeux des internautes curieux qui se sont retrouvés à y contribuer au tout début des années 2000, comme un territoire vierge, un nouveau projet faisant prendre une dimension supplémentaire aux principes du logiciel libre tel qu'il était connu. Ce sentiment de terre vierge se rapproche de la notion de « frontière électronique », qui symbolise l'immensité des possibilités offertes par Internet à ses créateurs et premiers utilisateurs et l'absence de normes et de contraintes. Fréquemment et avec humour, certains contributeurs de Wikipédia se considèrent comme « hackers de la culture ».

Sans forcément prévoir à l'époque le formidable succès actuel de Wikipédia, on sent dans les réponses de ces « pionniers » tout l'enthousiasme d'un projet qui leur a rapidement donné envie de s'investir.

RecentChanges

HomePage | RecentChanges | Préférences

Modifications depuis les 90 derniers jours

1 jour | 3 jours | 7 jours | 30 jours | 90 jours
 Afficher les dernières modifications partir de November 15, 2001 7:09 am

November 15, 2001

- (diff) Télécommunications 7:09 am (2 modifications) [Liste des inventions] Valéry Beaud
- (diff) FAQ 4:42 am (8 modifications) [Suite de la traduction] Buzz

November 14, 2001

- (diff) HomePage 5:04 am (8 modifications) [Ajout de l'index des pages (on passe la centaine !!)] Buzz

November 13, 2001

- (diff) Bienvenus, Nouveaux Venus 7:15 am (2 modifications) [La traduction continue] Valéry Beaud

Page des modifications récentes en novembre 2001

3. Des règles de vie au sein d'un des plus grands projets collaboratifs du monde

Comme on l'a vu précédemment, la communauté des contributeurs à l'origine de Wikipédia est très marquée par le logiciel libre. Elle est d'ailleurs composée en grande partie de personnes ayant une pratique quotidienne de l'informatique. Assez logiquement, le mode de fonctionnement de Wikipédia se calque donc sur les pratiques des développeurs informatiques et des forums informatiques alors très présents sur Internet.

Petit à petit, au gré des contributions bonnes ou mauvaises, des arrivées et des départs de contributeurs, du développement général de la communauté et du contenu, des règles de fonctionnement apparaissent pour structurer le travail collaboratif et les relations humaines. De par l'origine des premiers contributeurs, le vocabulaire et les fonctions de base de cette organisation communautaire se calquent sur celle du monde informatique : le « sysop » (administrateur système), appelé « administrateur » sur la Wikipédia francophone, en est une des figures principales.

La communauté fait face à ses premières attaques (appelées « vandalismes ») en les signalant et en listant les blocages d'adresse IP qui sont effectués. Mais les choses se font très progressivement, par exemple entre le 9 juin 2002 et le 31 août 2002 ne sont signalés que 3 vandalismes sur

la page de coordination ¹. Les vandalismes par robot inquiètent une communauté réduite qui craint de se faire submerger par des attaques trop massives.

Les premiers administrateurs, chargés de supprimer les pages ne correspondant pas aux critères encyclopédiques, ainsi que de bloquer les contributeurs malfaisants, sont nommés en 2002 sur la Wikipédia francophone. Auparavant, c'est Brion Vibber, développeur de MediaWiki, qui s'occupe de la maintenance, sur demande des contributeurs. Avec le changement de logiciel de Wikipédia fin 2002, un début de procédure apparaît pour nommer des administrateurs : il faut parler anglais, s'inscrire sur les listes de discussion anglophone et c'est Jimmy Wales qui décide de la nomination ou non d'un administrateur. Du côté communautaire, les bases de ce qui est demandé à un administrateur sont posées rapidement, et ne bougeront plus réellement jusqu'à aujourd'hui : être un contributeur actif et de confiance, comprendre que ce statut est avant tout technique et non éditorial, et qu'il est fait pour servir le contenu de Wikipédia et la communauté. Florence Devouard (Anthere) écrit sur la page [[Wikipédia :Administrateur]] en novembre 2002 :

Il est important de signaler que l'administrateur n'a en aucun cas davantage de pouvoir que les non-administrateurs en matière de création, d'édition, de forme, de contenu, de ligne directrice. Un administrateur doit être au service d'une meilleure fluidité de fonctionnement des pages et de leurs rapports entre elles, et à un rôle de protecteur de l'information contre le vandalisme. C'est tout.

Cette description de l'administrateur, qui est toujours en vigueur sur Wikipédia, ne doit cependant pas occulter une réalité plus nuancée : la confiance qu'inspire l'administrateur lorsqu'il postule, basée sur la qualité de ses contributions, la quantité de celles-ci (le nombre de contributions jugées habituellement nécessaires pour postuler n'a cessé d'augmenter depuis les origines, atteignant en 2012 plus de 10 000) et ses compétences à gérer les relations humaines, se transforme une fois élu en une certaine autorité, d'ailleurs souvent davantage perçue comme telle par les non-administrateurs que par les administrateurs. Même si l'administrateur donne son avis de pur contributeur sur un sujet encyclopédique, il

1. Page [[Wikipédia :Vandalisme en cours]], 31 août 2002.

sera très souvent vu d'abord comme un administrateur, avec les dérives autoritaires ou anti-autoritaires qui peuvent parfois en découler. Chez certains contributeurs, l'accession au statut d'administrateur tient également lieu de reconnaissance communautaire forte. Tout un vocabulaire folklorique entoure le statut de l'administrateur, souvent ironique (par exemple le balai, symbole des administrateurs qui nettoient l'encyclopédie de ses scories).

L'habitude de fournir des archives, des logs des actions effectuées, est présente sur Wikipédia dès son démarrage, comme dans beaucoup de projets informatiques. Les pages de log des suppressions de pages sont accessibles depuis octobre 2002¹.

Comme dans toute communauté sur Internet, les contributeurs de Wikipédia se dotent d'outils de communication dans le courant de l'année 2002 : listes de discussion, page de forum sur Wikipédia (rapidement appelée le Bistro).

Il faut noter que l'ensemble des règles de bases de Wikipédia s'est formé très rapidement après l'ouverture du projet, et qu'elles ont été formalisées sans réellement être remises en question jusqu'à aujourd'hui. L'utopie d'un fonctionnement sans règle ou « démocratique » n'a jamais été réellement présente sur Wikipédia, même si la liberté individuelle y tient une place très importante. Deux pages communautaires symbolisent à la fois cette grande liberté et les exigences minimales de fonctionnement de Wikipédia : la page des « principes fondateurs », qui expose en cinq points les règles intangibles de Wikipédia², et la page intitulée « Ce que Wikipédia n'est pas »³, qui expose en négatif les écueils que l'encyclopédie doit absolument éviter tomber, laissant pour le reste une grande liberté aux contributeurs (on notera particulièrement le point « Wikipédia n'est pas une démocratie »).

Sur la version anglophone, les règles sont écrites dans le courant de l'année 2001. Elles sont formalisées sur la version francophone dans les mois qui suivent.

1. Par exemple, la page [[Wikipedia :Deletion]] log octobre 2002.

2. http://fr.wikipedia.org/wiki/Wikipédia:Principes_fondateurs.

3. http://fr.wikipedia.org/wiki/Wikipédia:Ce_que_Wikipédia_n'est_pas (page créée en novembre 2001).

Les règles de fonctionnement de Wikipédia ne vont ensuite plus réellement changer, du moins sur le fond. La forme, elle, se complexifie de plus en plus au fur et à mesure des années, menant vers une sorte de bureaucratie touchant tout ce qui porte sur les prises de décisions et choix éditoriaux généraux, qu'il s'agisse des choix concernant l'importance et la présentation des sources, les règles typographiques ou de mise en page, mais également les questions touchant au droit d'auteur et plus généralement tout ce qui concerne la vie en communauté.

C'est ainsi que des instances de médiation se mettent en place, non pour juger du contenu des articles, mais des rapports entre contributeurs (Comité d'Arbitrage, « Wikipompiers », salon de médiation). Fidèle à son exigence d'ouverture et de neutralité de point de vue, ce refus de juger du contenu hors de la question de sa présentation ou des sources utilisées, entraîne une recherche permanente du consensus et du respect des règles de savoir-vivre qui donnent une certaine lourdeur aux décisions réglementaires de la communauté.

Tant que la communauté reste à taille « humaine », les rapports entre les contributeurs sont très resserrés, faits de beaucoup de proximité, mais également, par conséquence, de grandes implications personnelles pouvant mener à de spectaculaires conflits. Les premiers conflits graves entre contributeurs, rendant nécessaire une médiation formalisée, sont une sorte d'acte fondateur. Le Comité d'Arbitrage, fondé en 2005, consacre l'impossibilité d'une communauté consensuelle en permanence. Les décisions, la communauté s'étoffant, se parent d'un ensemble procédural lent, qui fait toujours référence au consensus comme à un idéal, mais provoquent également des discussions interminables, souvent génératrices de clivages et de demi-décisions.

Les divers moments de la vie communautaire se complexifient, comme les élections d'administrateurs, qui de quelques avis favorables déposés sur une page, deviennent des procédures codifiées où le candidat doit se présenter, justifier de son activité constante et sans éclats, de l'utilité qu'il aurait du statut d'administrateur, et recueillir un pourcentage de voix très largement majoritaire (environ 80% de votes favorables) pour espérer acquérir ce statut.

Atteignant à partir de 2007-2008 un chiffre (qui reste stable) d'environ 1 000 contributeurs très actifs, pour environ 15 000 contributeurs ac-

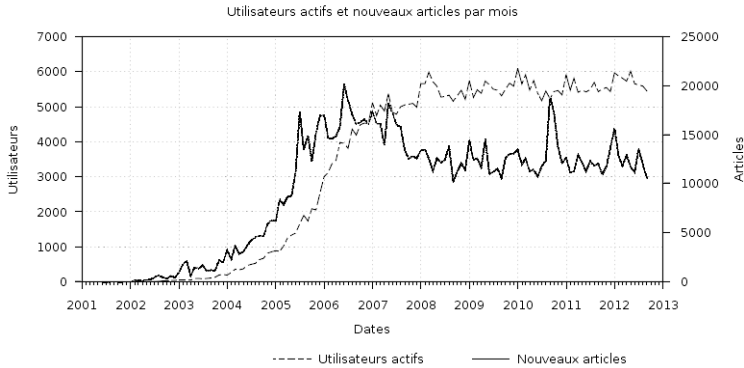
tifs chaque mois dans la version francophone, la communauté n'est plus aussi familiale qu'aux débuts de Wikipédia, et s'est structurée en différents projets recouvrant des thématiques spécifiques (Histoire, Astronomie, Cinéma etc.), lesquels ont un rythme d'activité variable, selon les contributeurs qui y sont attachés. Ces projets regroupent les contributeurs par centres d'intérêt, leur offrant un espace de discussion sur les sujets encyclopédiques, ainsi qu'un outil de coordination pour la rédaction.

De manière générale, l'organisation de la communauté des contributeurs de Wikipédia est assez légère : si Wikipédia en tant que telle est une œuvre intrinsèquement collaborative, elle n'est pour autant pas intégralement collective. Chaque contributeur reste bien identifié, chaque contribution faite est inscrite dans les historiques des articles et des pages de discussion, ou dans les espaces communautaires. Cette individualisation des contributeurs favorise une organisation horizontale où sur le principe seuls comptent les arguments éditoriaux et les sources pour résoudre un conflit, et non une quelconque autorité éditoriale, même si concrètement la « réputation » d'un contributeur, la confiance que le reste de la communauté lui accorde, implicitement ou en lui ayant confié un mandat d'administrateur ou d'arbitre, recrée une sorte de hiérarchie implicite et incompréhensible pour les personnes extérieures à la communauté. Ce sentiment de hiérarchie invisible, souvent nommée « cabale » par les contributeurs qui la dénoncent, n'empêche pas la création de rapports de force très forts au sein même des habitués. Entre « inclusionnistes » et « suppressionnistes » qui voient de façon extrêmement différente l'ouverture que doit avoir Wikipédia pour de nouveaux articles, par exemple, les éventuels conflits se passent sur des lignes quasi invisibles pour un œil extérieur à la communauté, et toujours sans aucune autre régulation que le dialogue et le respect des règles, ainsi qu'une exigence de consensus entre contributeurs et administrateurs pour exclure un contributeur problématique.

4. Wikipédia, un projet libre au-delà du Libre

Une dizaine d'années après sa naissance, que reste-t-il de cette genèse, de cette généalogie libriste de Wikipédia et des wikipédiens ?

Le nombre de contributeurs sur Wikipédia a connu une brutale augmentation à partir de 2005-2006. Le projet est devenu plus connu, l'arrivée



Utilisateurs actifs et nouveaux articles par mois sur Wikipédia en français. Par Akeron (CC-BY-SA 3.0, Wikimedia Commons).

massive de l'Internet haut-débit dans les foyers et la naissance du Web contributif et collaboratif (le Web 2.0, incarné à cette époque par les blogs notamment) a permis également au grand public de lire et de contribuer à Wikipédia. Par conséquent, les nouveaux contributeurs représentent une plus grande diversité que précédemment.

De projet identifié comme « geek », « pour informaticiens », Wikipédia est devenu un site « grand public », consulté par une part grandissante des internautes et donc écrit par des contributeurs venus d'horizons plus diversifiés. Beaucoup des contributeurs arrivés sur Wikipédia à cette époque et jusqu'à aujourd'hui ne perçoivent pas immédiatement le caractère « libre » du projet. Wikipédia est d'abord perçue comme un site de partage de connaissances, avant que les contributeurs (et encore moins les lecteurs) ne comprennent ce que signifie la licence qui régit les contributions et les conditions de réutilisation.

Bien des discussions au sein de l'encyclopédie font état de cette méconnaissance des termes des licences libres. Au-delà même, le développement des projets, et notamment la montée en puissance de Wikimedia Commons, la médiathèque libre qui accueille à partir de 2004 l'ensemble des illustrations des différentes Wikipédia, pose des questions sur le droit d'auteur aux contributeurs de Wikipédia : si écrire des textes et les placer volontairement sous licence libre n'est pas quelque chose de très com-

pliqué, illustrer des articles avec des images également librement réutilisables pose des questions plus complexes. Cela signifie que certains sujets, s'il est impossible de leur trouver des illustrations dans le domaine public ou sous licence libre, ne seront pas illustrés. Les différences culturelles sur le droit d'auteur sont également sources de confusion, notamment à cause du *Fair Use* pratiqué par les Anglo-Saxons sur la Wikipédia en anglais, qui facilite l'illustration mais ne peut raisonnablement s'appliquer dans un contexte de droit d'auteur européen (et notamment français). L'usage du *Fair Use* est explicitement refusé par la communauté francophone en 2006¹.

Mais des discussions reviennent régulièrement sur la question de la clause de réutilisation qui permet une réutilisation très libre des contenus, y compris pour un usage commercial : compliquant l'inclusion de contenus lorsque des institutions ou des contributeurs ne souhaitent pas voir leurs contenus utilisés par tous et pour n'importe quel usage, cette clause est régulièrement rediscutée et mise en cause. C'est alors qu'interviennent en général les contributeurs familiers avec les principes et les valeurs du Libre, qui rappellent, inlassablement, quels en sont les fondements et pourquoi une réutilisation limitée à un usage non-commercial ne correspond pas à la philosophie de Wikipédia, en ce qu'elle bride une grande part des possibilités de diffusion et de réutilisation des contenus.

À titre d'exemple, une discussion sur le « Bistro » de Wikipédia (espace central de discussion pour les contributeurs, découpé en journées) en octobre 2005 fait intervenir un contributeur qui se plaint de la complication des relations avec les institutions culturelles qui pourraient diffuser du contenu sur Wikipédia à cause de la possible réutilisation commerciale, et propose de « redéfinir les relations de Wikipédia avec le commercial, s[m]on opinion à ce sujet étant l'interdiction d'une réutilisation commerciale autre que très partielle de l'Encyclopédie ». Immédiatement, des contributeurs lui rappellent pourquoi la licence utilisée sur Wikipédia (à l'époque la GFDL) a été choisie et font référence au logiciel libre :

— Une restriction sur ce sujet risque à mon avis de limiter le développement de Wikipédia. Je n'imagine pas une seconde que Linux soit arrivé où il en est aujourd'hui avec une telle restriction.

1. Voir « Prise de décision : interdiction du Fair Use », septembre – octobre 2006 http://fr.wikipedia.org/wiki/Wikipédia:Prise_de_décision/Interdiction_du_fair_use.

— L'idée phare de Wikipédia, celle qui est la plus importante, je n'ai pas peur de le dire, c'est de produire du contenu libre.

— Surtout, obtenir une exception pour Wikipédia continue d'empêcher la libre reproduction hors de Wikipédia, donc ça ne rend pas Wikipédia libre, donc je ne vois pas à quoi ça sert.¹

Ces discussions ne se sont pas réellement arrêtées depuis cette époque, et le sujet revient très régulièrement dans les discussions entre contributeurs. Certains n'acceptent pas réellement l'intransigeance libriste des principes de Wikipédia et, s'ils continuent à contribuer par envie de partager leurs connaissances, n'adhèrent pas totalement à la philosophie du projet.

En 2008-2009, après des années de contribution sous licence GFDL, la Wikimedia Foundation prend la décision de faire basculer Wikipédia sous une licence plus compatible avec le contenu textuel de l'encyclopédie et permettant une réutilisation plus aisée, la licence Creative Commons BY-SA. Mais cette opération n'est pas facile, un tel changement de licence ne pouvant théoriquement se faire qu'après un consentement explicite de chaque contributeur, chose évidemment impossible à obtenir après 8 ans d'existence et des millions de modifications par des dizaines de milliers de contributeurs. Après conseils de juristes et de la Free Software Foundation, une grande consultation des contributeurs est lancée, afin de recueillir un nombre d'avis suffisamment significatif sur ce choix de changement de licence². Les contributions précédant ce changement étant doublement licenciées en GFDL et Creative Commons BY-SA.

Ce changement de licence facilite la compréhension des conditions d'utilisation de Wikipédia, à la fois pour ses lecteurs et ses contributeurs. Depuis, la Wikimedia Foundation travaille régulièrement à une simplification des explications concernant les licences, afin que la réutilisation des contenus et la contribution soient faites en toute connaissance de cause et que la pédagogie sur les licences libres soit de plus en plus comprise. Ainsi en 2012, toujours après un processus communautaire, la rédaction des conditions d'utilisation des projets Wikimédia a été reprise totalement et simplifiée.

1. Bistro du 19 octobre 2005. Le « Bistro » est le forum de discussion généraliste de Wikipédia, lieu de discussion principal des contributeurs.

2. Explication de la procédure de changement de licence sur [meta.wikimedia.org](http://meta.wikimedia.org/wiki/Licensing_update/fr)
http://meta.wikimedia.org/wiki/Licensing_update/fr.

Il y a donc coexistence à l'intérieur de la communauté wikipédienne entre des contributeurs très au fait des réalités et des valeurs du Libre, et de contributeurs qui soit n'en ont pas réellement conscience, soit ne voient pas cela comme très important. Ils contribuent sur un projet libre sans être libristes, et si beaucoup découvrent par Wikipédia le Libre dans sa multiplicité, de nombreux contributeurs réguliers de Wikipédia ne voient les licences libres que comme un cadre de contribution, et non comme une nécessité. L'important pour eux est Wikipédia, le site internet et ce qu'ils y écrivent, et non tout l'appareil de licences et de valeurs qui sous-tendent le projet.

Wikipédia rassemble donc bien au-delà de la communauté libriste des contributeurs qui en partagent un grand nombre de valeurs mais ne s'en sentent pas forcément proches.

5. Wikipédia, un projet politique ?

Depuis quelques années, les propositions de législation autour d'Internet et de la question des droits d'auteur reviennent souvent et certaines d'entre elles peuvent potentiellement impacter le fonctionnement de Wikipédia. Se pose donc régulièrement la question, chez les contributeurs, de la pertinence d'une action contre certains projets de lois afin de préserver le fonctionnement du projet sur lequel ils contribuent. En d'autres termes, quelle doit être l'implication politique de Wikipédia et de ses contributeurs.

Ces questions se posent davantage en termes de libertés sur Internet que de droit d'auteur, même si les législations sur le droit d'auteur posent fréquemment des limites à la contribution sur les projets Wikimedia (par exemple l'absence de liberté de panorama en France, qui interdit la diffusion sous licence libre des œuvres architecturales sous droit d'auteur construites en France, y compris lorsqu'elles sont financées par de l'argent public et placées dans l'espace public). Sur la question du droit d'auteur, il est relativement admis au sein de la communauté que ce droit d'auteur doit être respecté, quoi qu'on en pense en son for intérieur, et la chasse aux « copyvio » (mot-valise décrivant les « violations de copyright » dans les projets Wikimedia, telles que photos ou textes non-libres,

recopiés par des contributeurs) est très active, notamment sur Wikipédia et Wikimedia Commons. Cela étant, des tiraillements sont perceptibles sur certains points, notamment la question des photographies d'architecture récente dans l'espace public (liberté de panorama), où les contributeurs francophones se sont prononcés pour une « exception » au droit d'auteur, ou l'acceptation de photographies de monnaies et timbres¹. D'autres contributeurs se prononcent pour une licence de diffusion libre refusant la réutilisation commerciale (de type CC-BY-NC), afin de favoriser l'insertion notamment d'illustrations (de nombreuses institutions se disant beaucoup plus ouvertes à une réutilisation de leurs contenus sur Wikipédia si la réutilisation commerciale n'était pas autorisée). Ces contributeurs estiment travailler à une Wikipédia la plus complète (donc illustrée) possible, et ne se sentent pas forcément concernés par la partie « diffusion » et « réutilisation » du projet (deux caractéristiques qui sont pourtant fondamentales dans le monde du Libre).

C'est ainsi qu'en 2011 et 2012, la communauté s'est interrogée sur l'action à avoir face à des propositions législatives considérées comme attentatoires à la liberté d'expression et au fonctionnement de Wikipédia. La communauté italophone a entamé les actions, à l'automne 2011, avec son black-out en réaction à un projet de loi italien concernant la diffamation et jugé par les contributeurs comme étant extrêmement dangereux pour l'un des principes fondateurs de Wikipédia, la neutralité de point de vue. Après plus de 24h de black-out, empêchant tout internaute d'accéder à la Wikipédia en italien et le renvoyant vers une explication du « Comma 27 » et de ses conséquences potentielles, un retentissement considérable sur les réseaux sociaux et la presse, le projet a été ajourné (il est aujourd'hui quasi enterré) et la communauté s'est donc conduite comme un groupe de pression contre un projet de loi².

Cette première action collective contre une mesure législative, au nom de Wikipédia, a été ressentie comme un séisme au sein de la communauté des contributeurs. La communauté, notamment sur la version francophone, s'est partagée assez sensiblement entre un soutien aux contributeurs italiens et une réprobation de l'utilisation de l'accès à Wikipé-

1. Voir https://fr.wikipedia.org/wiki/Wikipédia:Exceptions_au_droit_d'auteur.

2. Sur le black-out de la Wikipédia en italien, voir les analyses de Pierre-Carl Langlais sur son blog Wikitrekk : <http://wikitrekk.blogspot.fr/2011/10/wikipediait-sur-le-point-de-disparaitre.html>.

dia comme moyen de pression politique. Si globalement l'ensemble des contributeurs réguliers reconnaissent l'intérêt des licences libres et les nombreux avantages qu'elles apportent à la construction de Wikipédia, de même qu'ils regrettent les limitations à la réutilisation des contenus (surtout iconographiques) qu'impose le droit d'auteur actuel, ils ne sont pas d'accord forcément pour passer au-dessus des lois (ce qui pourrait mettre en péril l'existence même de Wikipédia) ou même pour militer, laissant cela à la Wikimedia Foundation, hébergeur de Wikipédia, ou aux associations locales de soutien aux projets Wikimédia, comme l'est en France l'association Wikimédia France ¹.

Le même clivage s'est reproduit en janvier 2012 au moment de l'opposition d'une grande part des sites Web (essentiellement américains) aux projets de loi SOPA et PIPA. Cette fois-ci, la Wikimedia Foundation, après s'être officiellement prononcée contre ces lois, a organisé une large consultation de la communauté de la Wikipedia anglophone avant de mettre en place un black-out sur l'ensemble de la version anglophone le 18 janvier 2012, black-out également mis en place, de façon plus ou moins totale, sur un grand nombre de versions linguistiques. La communauté anglophone avait discuté pendant plus d'un mois auparavant pour savoir quelle attitude adopter. La communauté francophone, elle, s'est assez violemment déchirée sur la question : les partisans d'une action se référant à nouveau aux origines libristes de Wikipédia pour estimer que Wikipédia était, de par sa nature, un objet politique et qu'il était donc normal d'en défendre le fonctionnement et les conditions d'élaboration. D'un autre côté, une autre partie de la communauté estimait que Wikipédia devait rester absolument neutre, et ne pas prendre parti dans des questions politiques. Cette exigence de neutralité se base sur un des cinq principes fondateurs de Wikipédia, qui invite à la neutralité de point de vue dans les articles (même si cet idéal de neutralité est parfois difficile à comprendre et à atteindre, puisqu'il demande un équilibre savant dans les sources et la rédaction de l'article afin de refléter de façon juste l'état des connaissances sur un sujet, en y intégrant les controverses). Les opposants à toute action

1. Il faut noter que l'association Wikimédia France s'appelle officiellement « Wikimédia France – Association pour le libre partage de la connaissance », ce qui lui donne un objet social plus large que le seul soutien des projets Wikimédia et lui permet de soutenir de manière plus générale les actions en faveur de la connaissance libre (au sens des licences libres).

politique de la communauté wikipédienne justifient leur refus par le fait qu'il serait impossible, après une prise de position politique marquée de la part du mouvement wikimédien, de se proclamer neutre. Ce à quoi les partisans d'une action collective au nom de Wikipédia en cas de danger législatif répondent que le projet de Wikipédia lui-même est politique, et donc non neutre, si les articles encyclopédiques doivent l'être :

Mais non, la neutralité c'est pour le contenu encyclopédique, pas pour Wikipédia. Wikipédia a toujours été favorable au mouvement libre par exemple, c'est politique. Wikipédia a toujours été engagée pour la neutralité du net, c'est politique. Pour la liberté d'expression, en favorisant l'édition via des proxys anonymisant pour les citoyens des régimes répressifs. Tout ça c'est politique. La neutralité n'a aucun sens hors de l'espace encyclopédique, il est tout à fait légitime de mettre dans une page de discussion « j'ai mis ça dans l'article même si je pense que c'est faux », on fait tous ça tous les jours. Wikipédia a été fondé par des libristes et des humanistes (le site www.wikipedia.org n'est qu'une des facettes de Wikipédia, Jimbo a toujours mis le paquet également sur l'accès à l'information dans les pays du tiers-monde, etc.). Alors bien sûr, beaucoup de gens avec des profils différents y contribuent, mais le fonds de commerce n'a pas vraiment changé. Ton truc de neutralité, ça serait pareil que de demander aux restaurants du cœur d'être neutres politiquement, c'est stupide : les bénévoles s'engagent forcément dans un sens qui est cohérent avec leur démarche de bénévole. C'est pareil pour Wikipédia.¹

Le rapprochement entre défenseurs du Libre et défenseurs des libertés sur Internet est assez couramment fait, notamment face aux différents projets de « régulation » du Web liées aux questions de droits d'auteur (Hadopi, Acta, SOPA, PIPA par exemple). On retrouve dans les combats pour la neutralité ou la liberté du net de nombreux défenseurs des licences et des logiciels libres, qui partagent des convictions communes sur la fécondité de la décentralisation, de l'investissement personnel des internautes et un certain refus d'une soumission d'Internet aux intérêts de compagnies commerciales ou de sociétés de gestions de droits d'auteurs. La principale césure dans la communauté sur ces questions semble se faire

1. Arnaudus, bistro de Wikipédia, 17 janvier 2012.

entre les contributeurs qui estiment que Wikipédia, outre son aspect encyclopédique, porte dans ses fondements un projet politique, celui du libre partage des connaissances et de la construction collaborative du savoir, et ceux qui ne veulent pas mélanger leur volonté de partager leur savoir avec des considérations plus larges, voire politiques.

On sent donc un clivage certain entre les continuateurs de l'esprit libriste des débuts de Wikipédia et ceux qui ne se reconnaissent pas dans cette vision, sans que cela ait une quelconque traduction dans le contenu de l'encyclopédie : à part cette distinction, tous les contributeurs qui ont pris position sur ces sujets sont des contributeurs réguliers et tout à fait fidèles aux principes fondateurs de Wikipédia.

Est-ce une évolution de la communauté des contributeurs, ou même du projet Wikipédia ? Cette question est difficile : l'augmentation très importante du nombre de contributeurs, et de leur diversité, contribue certainement à d'importantes différences dans la manière d'envisager sa participation personnelle à un tel projet et à ce qu'il peut représenter. Certains estiment que leur contribution à Wikipédia, par exemple, est de l'ordre du loisir pur, qui n'a que peu à voir avec les licences libres ou la défense d'une certaine vision d'Internet et de la circulation des connaissances : il est difficile de savoir quelle est leur vision du projet, mis à part le fait que toute interprétation « politique » est refusée. Il faut également mesurer l'importance des contributeurs qui viennent sur Wikipédia pour rédiger un ou plusieurs articles sur des thématiques bien précises, qui les intéressent à divers degrés et pour des raisons variées (parmi lesquelles peut rentrer la « promotion » d'un sujet) et qui ne s'insèrent pas dans la communauté, dans le projet Wikipédia. De ceux-là il est impossible de connaître les valeurs ni le rapprochement qui pourrait être fait avec les valeurs du monde du Libre, même si on peut supposer que beaucoup d'entre eux ignorent ces connexions et sans doute même probablement ce qu'est réellement une licence libre.

Mais il est probable que cette question de l'engagement visible ait pu également porter des divisions quelques années plus tôt, alors même que les contributeurs étaient moins nombreux et plus homogènes dans leurs profils. Cela tient sans doute au caractère non organisé de la communauté des contributeurs : sans responsable éditorial, sans organisation préexistante, chaque engagement ou décision importante doit être longuement

débatte, discutée, afin de trouver un consensus satisfaisant la plus grande part de la communauté. Étant entendu que comme pour tout projet collaboratif, la « communauté » qui s'exprime n'est qu'une part (plus ou moins importante) de la communauté réellement contributive, et qu'elle a en permanence des scrupules à prendre des décisions tranchées lorsqu'elles impactent bien au-delà des quelques centaines de personnes qui s'expriment dans les débats.

Conclusion

Wikipédia est-elle un projet emblématique de l'histoire du Libre ? Très certainement. C'est probablement le projet de contenu libre le plus important en taille. Probablement aussi une des communautés les plus grandes en nombre de contributeurs réguliers. La création et le succès de Wikipédia montrent qu'il est possible de construire un projet dynamique et de qualité reposant sur des licences libres et sur le travail collaboratif et bénévole.

Cependant, prétendre que les contributeurs de Wikipédia s'inscrivent tous dans le mouvement libriste et se reconnaissent comme libristes serait une erreur d'appréciation. Paradoxalement, ce qui est peut-être le projet libre le plus emblématique est majoritairement écrit par des contributeurs qui n'ont pas forcément conscience de faire partie de ce mouvement libriste et qui, pour certains, refusent même explicitement d'être inclus dans une démarche de militantisme libriste.

Si les principes du Libre sont dans l'ADN même de Wikipédia et présents dans chaque principe, chaque fondement de l'encyclopédie, on pourrait estimer qu'ils forment les racines d'un projet qui s'est étendu au-delà de ses bases et qui ne peut qu'amener davantage de nouvelles personnes au Libre. Ces principes sont si profondément ancrés dans les règles de fonctionnement de Wikipédia qu'il ne faudrait pas craindre que malgré une communauté qui ne se reconnaît pas majoritairement comme « libriste », le projet puisse dévier de ses grands principes fondateurs. Au contraire, il agit comme une porte d'entrée vers le monde du Libre pour un public varié et souvent néophyte, qui peut petit à petit mesurer la réalité et les valeurs d'un mouvement dont Wikipédia n'est qu'une facette.

Références

- AYERS, Phoebe, Charles MATTHEWS et Ben YATES, *How Wikipedia Works : And How You Can Be a Part of It*, San Francisco : No Starch Press, 2008.
- DEVOUARD, Florence et Guillaume PAUMIER, *Wikipédia : découvrir, utiliser, contribuer*, Grenoble : Presses Universitaires de Grenoble, 2009.
- FOGLIA, Marc, *Wikipédia média de la connaissance démocratique ? Quand le citoyen lambda devient encyclopédiste*, Limoges : Fyp, 2008.
- LIH, Andrew, *The Wikipedia revolution : How a Bunch of Nobodies Created the World's Greatest Encyclopedia*, New York : Hyperion, 2009.
- SANGER, Larry, « The Early History of Nupedia and Wikipedia : A Memoir », dans : *Open Sources 2.0 : The Continuing Evolution*, sous la dir. de Chris DIBONA, Mark STONE et Danese COOPER, Sebastopol : O'Reilly Media, 2005, p. 307–338.

Du logiciel libre pour l'Assemblée Nationale

Liberté du code *versus* liberté des usages

Jonathan CHIBOIS

*Seule la confrontation [avec les acteurs qui
doivent s'en saisir] réalise
ou irréalise l'objet technique.*

Madeleine Akrich (1987, p. 208)

Quand, fin 2006, l'Assemblée nationale annonce son choix de doter pour la treizième législature, en juin de l'année suivante, de logiciels exclusivement libres les postes informatiques des bureaux parlementaires du Palais Bourbon, l'histoire de l'informatique en France paraît prendre un tournant. À cette époque, l'alternative libre fait l'objet de considérations nouvelles de la part des administrations républicaines, en convainquant de sa maturité et de sa crédibilité à équiper non plus seulement les serveurs mais aussi les postes de travail de ses agents. C'est la Gendarmerie nationale qui, quelques mois auparavant, a initié le mouvement en choisissant de déployer à grande échelle la suite bureautique libre OpenOffice.org¹, dans les environnements logiciels de type Windows. Quand l'Assemblée

1. Anthony MILAN, « Migration de la gendarmerie sous OOo », dans : *LinuxFr* (2005), URL : <http://linuxfr.org/news/migration-de-la-gendarmerie-sous-ooo>.

nationale opte de son côté pour le « tout libre », le pas supplémentaire franchi apparaît comme décisif, puisque le logiciel libre semble alors quitter le champ restreint des utilisateurs technophiles précurseurs pour rencontrer la majorité des utilisateurs non-spécialistes pragmatiques¹. Cette annonce inattendue faite par l'Assemblée nationale a été largement commentée dans la presse spécialisée, jusqu'à en trouver des échos outre-Atlantique. Au premier rang, ce sont les défenseurs du mouvement qui se sont félicités de la nouvelle, y voyant les prémices de l'aboutissement de leurs combats, par la confirmation de la justesse de leur cause.

Pour autant qu'on puisse en juger aujourd'hui, si la bataille remportée était effectivement remarquable, le tournant n'a, lui, pas été aussi net qu'espéré, et peut-être cela n'est-il pas sans lien avec le fait que le projet lui-même n'a pas été à la hauteur des attentes initiales. Sa mise en œuvre difficile constitue un témoignage de la période très particulière qui fut celle du milieu de la décennie 2000, marquée par la poussée d'un sentiment d'insécurité économique, et techniquement en suspens du fait de la fin de vie annoncée de Windows XP et de l'attente impatiente de l'iPhone. Peut-être n'est-ce pas un hasard si le libre a été durant cette période mis sur le devant de la scène alors qu'un vide était laissé par les grandes firmes internationales de l'informatique.

Après plusieurs mois d'enquête passés à collecter des archives et des entretiens, je vais retracer ici l'évolution du projet de sa conception jusqu'à sa réalisation, puis à son déploiement auprès des utilisateurs. Assurément, l'histoire de la mise en œuvre du libre dans les postes informatiques des députés est singulière, en ce qu'elle montre les attentes qu'ont suscitées à cette période les logiciels libres, et donc les défis qui se sont imposés à eux. Plus largement, sera traitée ici – au travers de l'exemple de l'administration parlementaire – la manière dont l'État et la société civile s'organisent et s'adaptent aujourd'hui aux enjeux de l'information et de la communication. Ces réflexions doivent en outre être comprises comme une des facettes d'un plus vaste travail de recherche sur les tensions entre la technologie et le pouvoir au sein des pratiques parlementaires actuelles.

1. André H. CARON, Luc GIROUX et Sylvie DOUZOU, « Diffusion et adoption des nouvelles technologies : le micro-ordinateur domestique », dans : *Canadian Journal of Communication* 11.4 (1985), p. 369–389, URL : <http://www.cjc-online.ca/index.php/journal/article/view/403>.

Concernant le récit qui suit, quelques précisions s'imposent. En premier lieu, je me dois d'évoquer la complexité d'une telle reconstitution, qui doit moins à la patience nécessaire pour retrouver ses différents acteurs ou l'intrication de ces événements qu'à l'opacité qui les recouvre. L'institution exerce en effet une pression étonnante pour que soient préservées des regards les coulisses de la vie politique qu'elle accueille : en guise d'exemple, un délai de communicabilité de vingt-cinq ans s'impose pour ses archives administratives. Le temps aidant, cependant, les langues commencent à se délier, une reconstitution que l'on peut considérer comme hautement vraisemblable a pu être élaborée sur la base de témoignages recueillis auprès des anciens acteurs du projet. Si les événements rapportés ici prêtent peu à débat et font consensus parmi les interlocuteurs rencontrés, il est important de garder à l'esprit que ce récit est le résultat de la mise en perspective d'éléments subjectifs convergents destinés à restituer une version stabilisée des événements successifs. Aussi, cette histoire que je rapporte est nécessairement consensuelle, mon écriture gommant les points de vue de chacun des acteurs pour n'en garder qu'un. Sans s'engager ici dans une réflexion sur la valeur épistémologique d'un tel récit, il est néanmoins important de ne lire dans ce qui suit qu'une entreprise destinée à réunir en un seul lieu et support les fragments éclatés d'une histoire qui mérite qu'on s'y arrête. En définitive, le récit chronologique effectué ici, qui paraîtra peut-être frustrant à quelques-uns, n'est pas l'exacte réalité des événements passés mais en constitue l'hypothèse la plus probable, ce qui suffit déjà à soulever de nombreuses interrogations.

1. Genèse

1.1 Le Libre, instrument des stratégies d'intelligence économique

Il est nécessaire, pour commencer, de s'intéresser aux modalités selon lesquelles le logiciel libre a été introduit à l'Assemblée nationale, et pour cela de s'attarder un moment sur un personnage clé qui s'en est fait le vecteur, Bernard Carayon (UMP). Député depuis 1993, il est connu depuis la publication du retentissant rapport Martre¹ pour l'intérêt qu'il porte aux

1. Henri Martre, du Commissariat au Plan (aujourd'hui connu comme le Centre d'Analyse Stratégique), est l'auteur en 1994 d'un rapport intitulé : *Intelligence économique et*

questions de mondialisation et d'intelligence économique. Il est l'auteur de plusieurs rapports sur les enjeux relatifs à la sécurité de l'information des grandes entreprises françaises et à la mise en œuvre de stratégies compétitives à l'échelle internationale, où il plaide en faveur d'une politique de patriotisme économique. C'est en énumérant les menaces qui pèsent sur la France, notamment celles relatives aux technologies de l'information et de la communication, qu'il s'intéresse aux logiciels libres censés garantir une certaine sécurité du fait de la transparence des codes source. En développer l'usage est alors important pour préserver l'autonomie du pays, d'où la nécessité de soutenir avec volontarisme ce secteur économique stratégique¹. En décembre 2005, lors des débats autour du projet de loi relatif au droit d'auteur et aux droits voisins dans la société de l'information (DADVSI), Bernard Carayon s'illustre en pointant les enjeux économiques, technologiques et stratégiques que les questions de rémunération des auteurs et interprètes soulèvent. Refusant que soient assimilés le *gratuit qui rapporte à tous*, celui de la communauté du logiciel libre, et le *gratuit qui coûte à tous*, celui du piratage et de la copie, il attire l'attention sur le fait qu'un certain nombre d'articles de ce projet de loi ne doivent pas être maladroitement instrumentalisés contre le logiciel libre². Dans une seconde prise de position en juin 2006³, le logiciel libre constitue un prétexte pour plaider en faveur de l'interopérabilité des systèmes d'informations. Bernard Carayon récuse en effet l'idée qu'un groupe restreint d'acteurs économiques se voie donner toute légitimité pour verrouiller et contrôler l'accès à l'information sur Internet.

Alors que la loi DADVSI est fraîchement promulguée⁴, Bernard Carayon remet le 14 septembre 2006 au Premier ministre Dominique de Villepin un rapport dans lequel les technologies de l'information figurent comme un domaine stratégique d'action prioritaire, en raison de leur rôle

stratégie des entreprises (La Documentation Française), dans lequel est pointé le retard du dispositif d'intelligence économique français, confiné « dans une orientation résolument défensive, à l'heure où la compétition à l'œuvre sur les marchés globalisés appelle l'urgence d'une coordination des capacités offensives. » (p. 92, Conclusion)

1. Bernard CARAYON, *Rapport d'information sur la stratégie de sécurité économique nationale*, rap. tech. 1664, Paris : Assemblée Nationale, 2004.

2. *Questions au gouvernement*, compte-rendu intégral de l'Assemblée nationale, 21 décembre 2005.

3. Michel ROCARD et Bernard CARAYON, « L'encodage des œuvres numériques, un nouveau Big Brother ? », dans : *Le Figaro* (juin 2006).

4. La loi a été promulguée le 1^{er} août 2006 et publiée au Journal Officiel le 3 août 2006.

particulier dans la compétitivité globale européenne. En abordant les thématiques de la sécurité informatique, de l'espionnage, de la fracture numérique, est avancée l'idée d'une maîtrise nécessaire des technologies numériques par les États souverains. De fait, se forme ici le lien entre le patriotisme économique cher à Bernard Carayon et les notions d'interopérabilité, les standards ouverts et le logiciel libre. Ces derniers apparaissent logiquement comme des outils pertinents dans la lutte contre le monopole des intérêts privés d'acteurs économiques puissants, qui concurrencent les souverainetés économiques nationales¹. Ce message qu'il s'emploie à diffuser n'est pas différent de celui contenu dans la lettre que lui-même et Richard Cazenave (UMP), député de l'Isère, ont rédigée en février à destination de Jean-Louis Debré, alors président de l'Assemblée nationale, pour suggérer que les députés puissent avoir eux-mêmes le choix, à la rentrée parlementaire suivante, du système d'exploitation qui équipe leur poste de travail. Les débats autour du projet DADVSI ayant paradoxalement donné aux logiciels libres une visibilité nouvelle – en particulier aux yeux des députés – il était stratégique que l'institution parlementaire montre par l'exemple la faisabilité d'opter pour ces systèmes informatiques, avant que la cause ne tombe à nouveau dans l'oubli. Dans ce rapport, Bernard Carayon annonce publiquement le projet de permettre aux députés de travailler dans un avenir proche avec un environnement libre, et il y est d'ailleurs indiqué que le président de l'Assemblée nationale en a déjà approuvé le principe. Le projet semble de fait – en partie du moins – déjà sur les rails.

1.2 Quand rien ne s'oppose, le projet s'impose

Il faudra pourtant patienter jusqu'à la mi-novembre 2006 pour que l'institution annonce officiellement « la mise en œuvre de l'environnement logiciel libre des postes micro-informatiques des députés lors de la XIII^e législature »². Que s'est-il passé durant les deux mois qui séparent les deux annonces ? Le Bureau de la Questure et le président de l'Assemblée nationale ont commandité le cabinet ATOS-Origin pour étudier la faisabilité

1. Bernard CARAYON, *À armes égales. Rapport au Premier ministre*, Paris : La Documentation Française, 2006.

2. *Communiqué de la division de la presse*, Assemblée nationale, 22 novembre 2006.

du projet, les besoins des députés et les contraintes techniques de l'institution, ceci en partenariat avec le Service des Système d'Information (SSI) et des représentants de chaque groupe politique. Trois alternatives devaient à cette occasion être comparées : une solution entièrement basée sur Windows et sa suite logicielle associée Microsoft Office ; une autre basée sur Windows assortie néanmoins d'une suite de logiciels libres ; ou finalement un OS et un écosystème applicatif entièrement libres.

Auprès de Jean-Louis Debré, grande figure du gaullisme, ce sont les arguments relatifs à l'indépendance technologique de l'État souverain et à la sécurité nationale, mais aussi ceux plus humanistes de l'encouragement à la création d'un bien culturel commun et à la diffusion libre de la connaissance qui auraient été décisifs¹. Le soutien au secteur d'activité émergent que constituait le logiciel libre en France a aussi certainement pesé en sa faveur², puisqu'ainsi l'institution envoyait à l'économie un signal de confiance fort tout en prouvant sa fiabilité par l'exemple. La décision du président n'étant qu'un accord de principe, il restait à éprouver la faisabilité technique, humaine et financière du projet, ce qui fut fait deux mois plus tard quand ont été reçus les résultats de l'audit. Malheureusement, on ne saura presque rien de son contenu, sinon en creux. Vraisemblablement le projet n'a pas rencontré d'opposition majeure ni au sein des députés, dont un certain nombre de représentants ont contribué au processus de consultation, ni au SSI, qui a étroitement collaboré à la conduite de l'étude. Les quelques données ayant filtré indiquent que « les solutions libres offraient désormais des fonctionnalités adaptées aux besoins des députés et permettaient de réaliser de substantielles économies en dépit de certains coûts de mise en œuvre et de formation »³. En effet, Rudy Salles (Nouveau Centre), député des Alpes Maritimes, nous apprendra plus tard que les économies envisagées étaient de l'ordre 500 000 euros⁴. On comprend alors qu'en définitive, rien ne justifiait réellement de s'opposer au projet Carayon-Cazenave, le consensus qui entourait sa validation étant to-

1. Entretien téléphonique avec Bernard Carayon, le 31 juillet 2012.

2. Ainsi que le suggérait notamment Pascal BOULARD dans *L'Assemblée nationale choisit le logiciel libre*, *La Tribune.fr*, le 9 mars 2007.

3. Extrait cité par Christophe GUILLEMIN, « Linux : Ubuntu s'installe sur les PC des députés », dans : *ZDNet* (2007), URL : <http://www.zdnet.fr/actualites/linux-ubuntu-s-installe-sur-les-pc-des-deputes-39367717.htm>.

4. Interview de Rudy Salles, député de Nice, par *CIO Online*, 2008. URL : <http://www.cio-online.com/videos/lire-deputes-sous-linux-entretiens-273.html>.

tal. Le Bureau des Questeurs et le président de l'Assemblée ont alors opté pour celle des trois solutions qui correspondait le mieux à l'esprit initial du projet et offrait en même temps le maximum d'économies : celle du poste totalement libre¹.

1.3 Refonder l'infrastructure pour rénover les usages

Pour saisir entièrement le contexte, il faut maintenant expliquer qu'avant 2007, l'infrastructure réseau de l'Assemblée nationale était rudimentaire et, du fait d'une faible demande de la part des députés, l'équipement informatique fourni était minimaliste et seulement facultatif. En 2007, sur l'initiative de Jean-Louis Debré, l'institution a investi dans une profonde rénovation de son infrastructure réseau, condition nécessaire à un ambitieux projet de modernisation. À ce titre, il a été envisagé de doter les équipes parlementaires de deux ordinateurs, un premier pour l' élu et un second pour un collaborateur, ainsi que les outils minimaux nécessaires à leurs fonctions (suite bureautique, navigateur Internet, client de messagerie) et quelques services dématérialisés associés (bureau virtuel sur intranet, agenda électronique...). Chaque bureau parlementaire devait également bénéficier d'un réseau virtuel propre (soit 577 au total). Ainsi, de façon entièrement indépendante du logiciel libre, une grande partie de l'installation informatique à l'Assemblée a été, entre 2006 et 2007, remise à plat et reprise de zéro : il a fallu organiser le câblage de l'intégralité des locaux de l'institution, la mise en place d'un maillage réseau unifié entre les différents bâtiments, la mise en redondance de l'ensemble des équipements techniquement sensibles, puis la mise en place d'une solution d'authentification sécurisée au niveau utilisateurs². Ce long travail de modernisation a, en définitive, mis la majorité des députés devant le fait accompli, désormais contraints d'adapter leurs usages techniques aux outils nouvellement installés sur leurs bureaux.

Un élément important mérite encore d'être mentionné ici. Une fois la rénovation de l'infrastructure technique générale planifiée, le SSI a entre-

1. La solution mixte aurait en fait été tôt déconseillée par le SSI en raison de la complexité de sa mise en œuvre. Dans les faits, le choix semble alors avoir départagé le « tout Windows » et le « tout Linux ».

2. Avis d'attribution de marché public, *Maîtrise d'œuvre pour la sécurisation des systèmes informatiques de l'Assemblée nationale*, Bulletin Officiel des annonces de marchés publics, réf. 06-143520, annonce parue le 9 octobre 2006.

pris courant 2005 de préparer le nouvel OS devant prendre la relève d'un Windows XP vieillissant. Windows Vista, la version suivante développée par Microsoft, fut à ce moment, après différents tests et retours, jugé insatisfaisant¹ du fait de son état d'avancement insuffisant. Annoncé en 2001 et prévu pour 2004, cet OS a en effet connu de nombreux retards dus à un développement problématique, et c'est finalement en novembre 2006 que la version destinée aux entreprises est commercialement disponible, trop tard pour la planification prudente du SSI. Quant à la solution de préserver Windows XP, elle a été écartée tant l'imminence de la fin de sa commercialisation était claire, et au vu de la nécessité que l'OS bénéficie de la maintenance de son éditeur durant les cinq années de la législature à venir. De fait, dans cette situation d'impasse, on comprend comment la proposition Carayon-Cazenave a pu recevoir du côté du SSI un accueil attentif. S'il est donc clair que l'incertitude technique, fruit de la réflexion interne au sujet de l'OS, eut un rôle facilitateur dans le succès du projet Carayon-Cazenave, ce serait toutefois se méprendre que d'en déduire l'enthousiasme du SSI.

2. Réalisations

2.1 Deux missions contradictoires pour l'administration parlementaire

Le 22 novembre, date à laquelle est diffusé le communiqué de presse de l'Assemblée nationale officialisant le choix du libre pour les postes des députés, s'engage une nouvelle étape dans le processus. Le Bureau de la Questure donnant au SSI la mission d'en organiser la conception et la mise en place, et validant alors la concrétisation des réflexions des mois précédents, le projet incertain devient un dossier à mettre en œuvre. Il s'ensuit une période d'études et de planification, qui débouche début janvier de l'année suivante sur la publication au Journal Officiel d'un appel à marché public pour la « mise en œuvre de l'environnement logiciel libre des postes micro-informatiques des députés lors de la XIII^e législature »² et

1. Entretien avec un ancien administrateur du SSI, le 5 juin 2012.

2. Avis de marché public, *Mise en œuvre de l'environnement logiciel libre des postes micro-informatiques des députés lors de la XIII^e législature*, à Paris, réf. 07-373, Bulletin officiel des annonces de marchés publics, annonce parue le 4 janvier 2007.

de son cahier des charges inspiré des résultats de l'audit rendu par ATOS Origin. La position du SSI n'est dès lors ici pas anodine, et les enjeux qui l'entourent ne sont pas dénués d'intérêts. D'un côté ses prérogatives font de lui le garant du bon fonctionnement des ressources informatiques au service des députés, mais d'un autre il est de son ressort de gérer la maîtrise d'œuvre du chantier. De fait, le voilà pris entre deux exigences contradictoires : d'une part assumer le choix qui n'est pas le sien de bousculer les usages techniques des députés et d'autre part, de garantir à ces derniers qu'aucun ennui matériel ne vienne perturber le bon déroulement de leurs activités parlementaires.

Ainsi, il s'est très vite révélé crucial que tout ceci ne fasse que le minimum de remous, ce qui montre déjà que l'enjeu premier pour le SSI était relatif moins à la bonne facture du nouveau poste de travail qu'à sa réception par les utilisateurs. L'ergonomie utilisateur est devenue de fait son cheval de bataille, manière de concilier sa mission de service et celle de modernisation. Ce n'est pas le mécontentement des nouveaux utilisateurs que l'on craignait mais celui de ceux qui seraient nécessairement amenés à modifier leurs habitudes. Dès lors, il est apparu comme évident pour le SSI que les efforts devaient être mis sur un travail de réduction des écarts d'interface entre le système en place et le nouveau.

Début mars 2007, l'entreprise Linagora associée en partie à Unilog est désignée titulaire du marché, pour sa proposition peu coûteuse¹ de mettre en œuvre le populaire OS GNU/Linux Kubuntu de la société Canonical Ltd. L'avantage de Kubuntu résidait dans son interface déjà pensée pour le grand public, et présentait notamment certaines ressemblances avec les environnements de type Windows. Dans les prévisions de Linagora, Kubuntu constituait une base adéquate en ce qu'elle répondait déjà en tant que telle aux exigences avancées par l'Assemblée nationale en matière d'ergonomie pour les utilisateurs². Ce qui est apparu plus tard comme une erreur avait été pensé initialement comme une avance prise sur le chantier à venir. Au cours de ce mois de mars, les deux parties s'entendent sur un

1. Le budget de la proposition de Linagora était de près de 90 000 euros, pour la réalisation du poste de travail, les différentes sessions de formation auprès des administrateurs du SSI et des équipes parlementaires volontaires, et une année de maintenance après déploiement.

2. Entretien téléphonique avec Michel-Marie Maudet, directeur général adjoint de Linagora et chef du projet, le 6 septembre 2012.

calendrier prévisionnel resserré, les quatre mois restants pour la réalisation et le déploiement apparaissaient déjà courts pour un projet de cette envergure. Des rencontres régulières sont organisées au fur et à mesure de l'avancement du projet, ainsi que quelques ateliers avec des volontaires, notamment des collaborateurs parlementaires, pour recueillir avis et suggestions autant que pour désamorcer les inquiétudes naissantes chez les futurs utilisateurs. La fourniture du poste fonctionnel était prévue pour mi-juin, après une phase de tests intensifs.

2.2 Un outil libre est un outil souple

Ce n'est que quelques semaines plus tard, début avril 2007, que Linagora prend la mesure de la distance qui sépare l'attente réelle du client et la proposition commerciale pour laquelle ils se sont engagés. Par des exigences très poussées de personnalisation, le SSI se montre très pointilleux sur le poste de travail des députés, ce qui oblige Linagora à s'adapter en redistribuant ses ressources et en développant de nouvelles compétences : à cette époque est recruté un membre francophone de l'équipe de développement de Kubuntu¹, pour prendre en charge cet aspect inattendu du chantier. Le SSI a en effet vu son intérêt dans la liberté de modification du code des logiciels libres, c'est-à-dire que plutôt que d'imposer aux utilisateurs l'inconfort de s'adapter à une interface nouvelle, on allait faire en sorte que le système s'adapte, lui, aux utilisateurs. Des instructions en ce sens ont été explicitement rédigées dans la section *Ergonomie du poste de travail* du cahier des charges de l'appel d'offres : « le titulaire [du marché public] optimise l'ergonomie de la distribution afin de minimiser l'effort d'adaptation des utilisateurs à leur nouvel environnement. Cette optimisation doit porter sur la présentation générale du système mais aussi sur les automatismes acquis par les utilisateurs. (...) Un soin particulier doit également être apporté à la simplicité de l'interface présentée par le système »². Dans les faits, c'est l'interface visuelle de l'OS Windows 95 qui est recherchée pour sa sobriété et qui sera reproduite avec beaucoup de réalisme, ce qui rétrospectivement ne fut peut-être pas une stratégie payante, au vu de l'alourdissement de la charge de maintenance que cette personnalisation poussée a engendré les années suivantes.

1. Anthony Mercatante, en 2007 était *core developer* Ubuntu.

2. Cahier des Clauses Particulières, *Ibid.*

Néanmoins, on comprend qu'à cet instant, ce parti pris permettait de neutraliser efficacement les critiques des utilisateurs qui craignaient pour leurs habitudes d'usage. Dans le même sens, le SSI avait préalablement sélectionné les applications clés dont le futur OS devrait disposer : OpenOffice.org comme suite bureautique, Mozilla Firefox comme navigateur Web, Mozilla Thunderbird comme client de messagerie et VLC comme lecteur multimédia. Le but recherché ici était d'opter pour des outils qui avaient déjà fait leurs preuves, dont la maintenance à long terme par leurs éditeurs respectifs faisait peu de doute, et que les députés pourraient s'approprier sur leurs différents postes de travail, en circonscription notamment. D'évidence, le poste de travail des députés a donc été façonné sur mesure, de façon à prêter le moins possible le flanc à la controverse. En cela, la liberté et la souplesse offertes par les logiciels libres ont surtout été mises à profit dans une entreprise d'imitation de l'interface de l'OS préexistant.

2.3 Un projet retardé qui irrite

Ce qui ressort finalement de cette période est surtout la difficulté de communication entre Linagora et l'Assemblée nationale. Il est révélateur que le terme « client difficile » puisse être employé du côté prestataire pour qualifier l'Assemblée nationale par comparaison avec différentes expériences de contrat avec l'administration publique, et qu'il soit de plus reproché au SSI son « manque de sponsoring ». Ce qu'il faut entendre ici, ce n'est pas un refus de se soumettre à la demande du client, mais davantage l'expression d'un regret de travailler non pas *avec* mais *pour* un partenaire. Que le SSI considère, lui, que le prestataire n'a pas été à la hauteur, que les « limites de Linagora » ont été mises en évidence, illustre le même regret, quoiqu'en symétrie. Plus que la frustration de ne pas avoir trouvé chez ce partenaire une compréhension parfaite des enjeux spécifiques de l'Assemblée nationale, le reproche qui est ici adressé est qu'il n'ait pas pu mettre en œuvre les moyens nécessaires pour une entière réussite du projet, c'est-à-dire qu'il n'ait su ni correctement anticiper les difficultés ni s'y adapter. Quand d'un côté Linagora pointe la disproportion du projet au vu du temps et des moyens engagés, le SSI regrette pour sa part un manque de compétences techniques et de solidité structurelle chez son partenaire. Ces reproches mutuels sont peut-être moins des

rancœurs que les résidus de nombreuses appréhensions suscitées par un projet proprement expérimental.

En dépit de ces tensions, certainement inévitables dans la conduite d'un tel projet, le processus de développement du poste destiné aux députés a suivi son cours. Les semaines de développement se sont succédées et mi-juin, la maquette fonctionnelle a été livrée avec un mois de retard sur l'agenda prévu, ce qui s'explique par des ajouts de fonctionnalités en cours de réalisation. La phase de tests exhaustifs a suivi sur une durée d'environ un mois, occasion pour Linagora de se rendre sur le terrain pour constater les dysfonctionnements et tester ce qui n'avait encore pu l'être et ainsi finaliser le travail. Mi-juillet, près d'un mois après la prise de fonction des députés de la treizième législature, le nouveau poste des députés est terminé et intégré comme il se doit à l'infrastructure informatique de l'Assemblée nationale. La logistique aura de son côté quelques difficultés supplémentaires à tenir les délais, d'abord parce que les travaux parallèles de câblage des locaux n'étaient pas tout à fait clos, ensuite parce que la livraison assurée par une société tierce fut retardée. Début septembre, lors de la rentrée parlementaire, les postes de travail sont toujours en cours de déploiement, le SSI terminant de configurer le matériel tout juste livré. Au final, de la réalisation aux premières utilisations des députés, le chantier aura duré six mois. Le désordre engendré par ce retard, quoique moins conséquent de la mise en œuvre du logiciel libre que de la rénovation générale de l'infrastructure informatique, a provoqué une irritation générale chez les élus et collaborateurs, et a notablement contribué à l'émergence d'un mouvement de méfiance et de scepticisme vis-à-vis de la fiabilité de ce nouvel OS qui tardait à se mettre en place.

3. Confrontations

3.1 Premiers retours mitigés

Cette irritation n'a pas contribué à ce que soient relativisés les réels dysfonctionnements qui ont suivi, lors des premières tentatives d'utilisation dans le courant de l'automne. Si les différents protagonistes espéraient un horizon paisible à l'aube de la seconde partie du contrat, simple mainte-

nance¹ de la nouvelle configuration, leurs aspirations furent rapidement déçues à en croire la tempête qui s'est alors levée. Le premier motif des mécontentements rassemble toutes les difficultés rencontrées pour écouter de la musique et regarder des films, la lecture de DVD en est un exemple significatif. Les protections techniques (DRM) incluses sur ces supports destinées à empêcher toute utilisation abusive des œuvres culturelles, dont la valeur juridique a été instaurée par la nouvelle loi DADVSI², sont elles-mêmes protégées des regards curieux, condition indispensable pour éviter que quiconque puisse en découvrir le fonctionnement et ainsi passer outre. Or, un cadenas n'a plus de raison d'être si l'empreinte de sa clé n'a plus de secret, et la logique du logiciel libre est de fait incompatible avec celle de ces protections, ce que précisément dénonçait quelques mois auparavant le député Bernard Carayon. Quand bien même une application parviendrait à contourner ces protections dans une visée d'interopérabilité³, elle serait du fait de son caractère libre obligée d'en diffuser le mode d'emploi. Pour ces raisons, en dépit de la faisabilité technique, le SSI avait initialement pris le parti de retirer cette fonctionnalité aux lecteurs multimédias proposés sur le poste des députés, pour cause d'insécurité juridique⁴. Pour d'autres raisons de légalité mais avec des conséquences semblables, les formats audio MP3 et vidéo FLV, très répandus dans les usages connectés des députés, ont été également exclus du nouveau poste de travail. Les équipes parlementaires se sont donc trouvées dans l'impossibilité de lire des documents multimédia standards.

Le second grand dysfonctionnement posait un problème autrement épineux aux députés, non pas que l'importance d'un support d'information comme le document multimédia soit à prendre à la légère, mais ici ce sont leurs pratiques d'élus qui s'en trouvait bloquées. Le fait est que le nouveau système peinait significativement à interagir avec leurs smartphones,

1. « Le titulaire assure durant douze mois la maintenance adaptative et corrective de la configuration logicielle (...) ainsi que la mise à jour de la documentation fournie », *Cahier des Clauses Particulières*, *op. cit.*

2. Articles 331-6 et 331-7 du *Code de la propriété intellectuelle*.

3. Seule exception permise par la loi DADVSI qui autorise le contournement de ces protections, dans le but d'empêcher les pratiques abusives de restriction d'accès aux œuvres, de la part d'acteurs économiques trop zélés.

4. Si les défenseurs des logiciels libres ont pu espérer qu'un tel épisode sensibilise à leur cause les députés, il donne surtout raison aux craintes exprimées de Bernard Carayon durant les discussions autour de ce projet de loi, précédemment évoquées.

et au premier rang d'entre eux l'iPhone¹. Aucune synchronisation des données n'était possible pour la plupart des modèles sans manipulation complexe. En particulier, l'agenda ne pouvait plus être partagé entre les différents supports, et donc échangé entre les différents collaborateurs. Ici, la problématique n'était plus essentiellement juridique mais davantage commerciale, les constructeurs d'appareils mobiles n'ayant jusqu'ici trouvé que peu d'intérêt à investir dans la compatibilité de leur matériel avec les systèmes GNU/Linux. Pallier cela en lieu et place des constructeurs de smartphone était une tâche particulièrement chronophage, du fait de la grande diversité de produits déjà en circulation parmi les députés, et dont l'évolution du parc total suivait le rythme soutenu du marché. Cette entreprise a mobilisé une quantité très élevée de ressources de l'équipe dédiée à la maintenance du système, ce qui fut la priorité des premiers mois. Il n'aurait pas été illogique du point de vue technique que, dès le début de la nouvelle législature, l'Assemblée nationale impose un choix restreint d'appareils² préalablement testés pour leur compatibilité, mais c'était impensable d'un point de vue politique. Il a ainsi fallu redéployer certains systèmes Windows XP dans les bureaux parlementaires pour cette tâche unique.

Si les dysfonctionnements à cette époque, et également durant les mois suivants, ont pu paraître nombreux et importants au vu de l'ampleur des problématiques techniques soulevées, il faut néanmoins insister sur le fait que derrière une somme de mécontentements significativement relayés, une très grande majorité des députés ont sinon relativisé les difficultés, tout au moins pris patience. Le seul dysfonctionnement relevé de façon unanime fut l'application d'agenda partagé OBM proposée aux députés par l'intermédiaire de Linagora, qui n'a jamais été à la hauteur des espérances. L'alternative constituée par le très fonctionnel équivalent proposé par la firme Google vers lequel se sont finalement tournés les utilisateurs, a fait que ce problème, très vite, n'en est resté un que pour le SSI. En définitive, pour la grande majorité des députés la phase de tests en conditions réelles a effectivement permis d'évacuer l'essentiel des problèmes

1. Commercialisée en France, dans sa première version, en novembre 2007.

2. Ce qui a finalement été fait quelques mois plus tard, en guise de simple recommandation néanmoins. Il n'est de toute façon pas possible d'exiger des députés qu'ils s'adaptent aux seules contraintes de l'Assemblée nationale, aux dépens des autres mandats qu'ils exercent souvent en parallèle.

potentiels. La question est néanmoins toute différente pour la minorité des usagers les plus technophiles, c'est-à-dire les plus exigeants et les plus dépendants des outils numériques, et de fait les plus audibles.

3.2 Destinataires projetés, destinataires testés et destinataires effectifs

Compte tenu des polémiques soulevées, il émanait des députés l'attente d'un poste de travail multifonction, ce que n'était foncièrement pas le poste qui leur a été proposé. Cet écart significatif entre les besoins effectifs et les besoins auxquels Linagora a répondu, pose de façon centrale la question des modalités selon lesquelles ont été initialement définis les besoins de l'utilisateur. Avant d'entrer dans les détails, il est maintenant important pour y voir clair de préciser la terminologie car la figure de cet utilisateur est, de différentes façons, multiple. Comme nous allons le voir par la suite, le député (et derrière lui ses collaborateurs) n'est pas le seul utilisateur de son poste de travail, d'où la nécessité de le distinguer par une dénomination propre : le *destinataire*. En second lieu, c'est la figure elle-même du destinataire-député qui a évolué tout au long du projet, ce qui impose de dénommer ses manifestations successives.

Au cours de la phase de tests exhaustifs ayant eu lieu entre mi-juin et mi-juillet, il est apparu que les futurs destinataires, volontaires pour l'exercice, « ont remonté des problèmes qu'ils rencontraient, et qui étaient de vrais problèmes »¹ comme me l'a confié un des membres de l'équipe de réalisation. Que soit éprouvée ici la nécessité de qualifier de « vraies » les problématiques soulevées est déjà le signe d'un doute quant à la légitimité des fonctionnalités demandées par le SSI, mais surtout révélateur d'un fossé séparant le *destinataire-projeté* du *destinataire-effectif*². Cet écart entre les besoins envisagés durant la réalisation et les besoins ressentis par les utilisateurs, quoique certainement pas unique en son genre, attire l'attention sur l'organisation singulière des sphères d'activité des différents protagonistes. En somme, le SSI dans sa fonction de client de

1. Entretien avec Anthony Mercatante, le 5 juillet 2012.

2. Pour reprendre l'esprit de la terminologie de Madeleine Akrich. Voir Madeleine AKRICH, « Comment décrire les objets techniques ? », dans : *Techniques & Culture. Revue semestrielle d'anthropologie des techniques* 54-55 (1987), p. 205-219, URL : <http://tc.revues.org/4999>.

Linagora, joue un rôle important d'intermédiaire entre les futurs utilisateurs et les concepteurs qui ne peuvent se rencontrer. La phase de test en conditions réelles d'utilisation doit être vue alors comme l'occasion pour Linagora de dépasser ce fossé et de se confronter à un *destinataire-test*, c'est-à-dire à un usage enfin incarné qui faisait émerger des problématiques « vraies ». Il ne s'agissait plus de spéculer sur ce que le député devait pouvoir faire, ou d'anticiper les difficultés qu'il ne devait surtout pas rencontrer, ou encore d'aménager certaines facilités pour lui, mais au contraire de débloquer des situations concrètes. À l'issue de cette phase, on peut dire que les figures du *destinataire-projeté* et du *destinataire-test* se sont confondues en une seule, permettant de rapprocher significativement les besoins et les contraintes effectives d'utilisation.

Or, comme on l'a vu, la phase de tests n'a pas suffi à faire converger le poste de travail réalisé et le poste de travail attendu. De fait, le *destinataire-testé* et le *destinataire-effectif* ne se recouvrent que partiellement, le premier ayant manifesté (par l'intermédiaire du Bureau des Questeurs) son approbation pour le nouveau système à l'inverse du second (les députés irrités). Cette divergence s'explique vraisemblablement par le faible nombre de députés disponibles et volontaires pour accepter un mois durant l'inconfort d'intégrer dans la mécanique bien huilée de leurs pratiques quotidiennes de travail un outil non parfaitement finalisé, justement pour en éprouver les limites. Si on ne sait que peu de choses au sujet des *destinataires-testés*, ils n'étaient vraisemblablement que peu nombreux et certainement plutôt technophiles. Ceci éclaire déjà en grande partie le problème tant il y a de conceptions différentes du fonctionnement d'une équipe parlementaire, de ses prérogatives et ses outils. Si le témoignage ci-dessus amène à penser que la phase de tests en conditions réelles fut concluante, elle n'a cependant pas suffi à faire émerger les *destinataires-effectifs*. De fait, les *destinataires-testés* n'étaient dans leurs occupations et centres d'intérêt pas représentatifs de leurs collègues. Introduit dans le processus de développement en gage de substitut, le *destinataire-testé* devait valider l'adéquation du *destinataire-projeté* au *destinataire-effectif*, ceci sans pour autant mobiliser ce dernier, qui ne devait connaître aucun désagrément relatif à la conception du nouveau système. Ici, la figure qui illustre le mieux la position du SSI est celle du sas, garant de l'étanchéité de la cloison destinée à préserver les députés des désagréments liés à la

conception du poste, par l'introduction dans le processus du *destinataire-test*. Il n'est pas évident que cette stratégie ait été finalement couronnée de succès, vu l'ampleur des remous suscités par les désagréments malgré tout ressentis. Le SSI n'a effectivement pas été épargné par les critiques, ayant endossé l'entière responsabilité des dysfonctionnements et désagréments survenus¹, ce qui est autant dû aux responsabilités administratives qui sont les siennes devant le Bureau de l'Assemblée nationale qu'à sa position d'interlocuteur unique pour les députés sur les questions informatiques.

4. Usages

4.1 Mainteneurs-projetés, mainteneurs-effectifs

Si dans la première phase du chantier, seuls deux acteurs se trouvaient en présence, dans la seconde phase – celle de la maintenance à moyen terme du produit réalisé – ils sont désormais trois. Une redistribution des rôles s'opère en effet chez les différents acteurs, quand le poste est remis aux députés. En résumé, les députés-usagers n'ayant jusqu'ici qu'une existence fictive, devenue bien réelle pour les concepteurs et le maître d'œuvre à l'occasion de la confrontation, l'*utilisateur-effectif* entre ici en scène comme un nouvel acteur. À cette occasion, le SSI prend de son côté et comme prévu le rôle du bouc-émissaire. Notons que la position d'intermédiaire du SSI entre utilisateurs et techniciens demeure, bien que le sas n'ait plus de raison d'être, les députés ayant entre leurs mains le nouveau poste. Chez Linagora ensuite, un basculement survient quand l'équipe de conception laisse la place à celle de maintenance², engendrant cette situation intéressante où Linagora devient elle-même usager de son propre

1. Plusieurs témoignages recueillis remontent un certain sentiment d'injustice concernant cette situation, qui tout en étant inévitable et logique (« c'est comme ça ») fut source de souffrance de la part du personnel du SSI (« on a dû raser les murs, oui, pendant quelques mois »).

2. Bien que le terme employé au sein de l'entreprise soit le « support » pour désigner cette activité, j'utiliserai pour ma part celui de « maintenance » qui me paraît premièrement plus intelligible à la lecture et deuxièmement plus commode pour la suite de la description. En effet, ce choix me permet plus loin de mobiliser la figure du « mainteneur », terme certes peu répandu dans la langue française, mais particulièrement adéquat pour illustrer l'évolution fonctionnelle de celle du concepteur.

système. En fait, parmi les trois acteurs que sont Linagora, le SSI et les députés, deux d'entre eux sont désormais utilisateurs, et non un seul comme présagé jusqu'à présent, dont les usages respectifs sont fondamentalement différents tout en étant hautement complémentaires.

Face à l'utilisateur-député apparaît donc un utilisateur-mainteneur. En coulisses, ce dernier va regarder fonctionner la grande machinerie que constitue l'ensemble des postes des députés, de leurs réseaux privés et de l'infrastructure matérielle commune pour y apporter en temps réel les réglages nécessaires, remplacer les pièces s'usant prématurément dans l'engrenage ou ayant montré des signes tardifs de non-compatibilité avec l'ensemble. Sa tâche est de faire perdurer le système, envers et contre toutes les contraintes extérieures, qu'elles soient d'origine humaine ou technique. Condition nécessaire à la discrétion et à l'efficacité de ses interventions, ces dernières s'effectuent idéalement à distance par l'intermédiaire du réseau, à un moment où l'utilisateur-député ne pourra être importuné¹, et de manière automatisée de façon à faire autant que possible profiter des améliorations l'ensemble des 1 154 postes. Alors qu'était jusqu'à présent préservée à l'Assemblée nationale une équipe de techniciens d'intervention qui puisse se déplacer en relative urgence auprès des députés en difficulté, on imagine aisément l'intérêt que l'institution avait à demander que soit mise en place une telle organisation, pour les économies d'échelles d'une part et en ce que cela prolongeait la logique de dissimulation des contraintes matérielles aux députés d'autre part.

En dépit de fortes contraintes de sécurité propres à l'institution, ce fonctionnement centralisé d'intervention à distance a été effectivement mis en place, avec néanmoins quelques surprises techniques telles que l'exemple qui suit. Il faut néanmoins encore dire, pour bien saisir les enjeux qui suivent, que l'équipe de maintenance, tout en disposant d'outils spécifiques pour surveiller, diagnostiquer et intervenir, s'appuie sur les outils d'administration fournis par le système lui-même. Kubuntu comme Windows disposent notamment d'une base de données unifiée des paramètres de configuration pour une gestion efficace de l'ensemble de l'écosystème

1. Ceci a par ailleurs longtemps posé problème : pour effectuer une intervention à distance, il est opportun de patienter jusqu'à ce que le bureau parlementaire se vide, manière de s'assurer que les postes ne sont pas en cours d'utilisation, mais alors les postes sont éteints, ce qui rend toute manipulation impossible.

applicatif¹. Or, il est apparu que les applications principales – la suite bureautique, le navigateur Web et le client mail – intégrées au système sur la demande du SSI, n'étaient pas compatibles avec cette gestion unifiée des paramètres de configuration. Par conséquent, l'automatisation d'un certain nombre d'interventions, même bénignes², à l'échelle du parc informatique est devenue ardue, nécessitant de contourner beaucoup d'obstacles techniques successifs. Sans entrer dans les détails, le fait est que si du point de vue des députés ces applications tierces donnaient entière satisfaction, leur logique interne avait été conçue indépendamment de la base de données unifiée des paramètres de configuration. En résumé, l'exigence de proposer aux députés des applications reconnues et répandues s'est trouvée au sein du poste libre mise en opposition avec l'exigence d'une maintenance centralisée du parc³.

Cet exemple est significatif du fait que, pour la maintenance de même que pour les députés, il y a eu un net décalage entre les outils proposés et les besoins effectifs. Le mainteneur-projeté et le mainteneur-effectif entrent également, à l'instar des destinataires, en confrontation. Les problématiques futures qui devaient être celles de l'équipe de maintenance, telles qu'envisagées par les concepteurs, s'avèrent pareillement imprécises, si bien que les tâches du mainteneur-effectif sont finalement rendues difficiles. J'ai expliqué précédemment les écarts entre usages projetés et usages effectifs en évoquant la position d'intermédiaire du SSI entre concepteurs et destinataires qui n'a pas permis que se confrontent directement les différentes perceptions du destinataire type. Le problème est ici bien différent puisque la relation entre concepteurs et mainteneurs est remarquablement directe : Linagora endosse à elle seule les deux rôles. Comment dès lors expliquer que cette entreprise expérimentée n'ait pas

1. Pour intervenir sur la base de registre du système d'exploitation, on utilise sous Microsoft Windows l'application Regedit, et Kconfig sous Kubuntu. Je remercie les spécialistes de me pardonner la simplification outrancière du fonctionnement de ces deux OS, qui n'est malgré tout pas inexacte, pour la nécessaire clarté du récit.

2. Un exemple dont je dispose fait état de la page d'accueil du navigateur Mozilla Firefox, dont seule l'URL devait être modifiée suite à une mise à jour de l'intranet à disposition des députés.

3. Le système Windows, associé à des outils comme Active Directory, est connu pour mieux gérer ce type de problématique propre aux postes de bureau. À l'inverse, dans une situation similaire sur des postes serveurs, l'architecture des systèmes GNU/Linux sera plus convaincante.

anticipé les usages de maintenance effectifs¹, alors qu'elle savait que la tâche lui reviendrait ? Il n'y a que peu de causes plausibles, et ce qui apparaît tend à montrer que la question de l'usage futur de l'équipe de maintenance ne s'est simplement pas posée², l'entreprise n'envisageant pas que la maintenance d'un poste de bureau puisse significativement différer de celle connue des serveurs. Encore une fois, il apparaît combien Linagora a été prise au dépourvu face à la demande de l'Assemblée nationale, chose peu surprenante du fait, comme on l'a vu, du caractère expérimental de ce chantier.

4.2 Adéquation des outils aux usages

Sur ce point précisément, il faut à ce stade du récit s'interroger plus profondément. Faut-il mettre l'ensemble des décalages constatés entre besoins et outils sur le compte de maladresses d'arbitrages dues aux inconnues qui constituaient le projet ? Parmi les éléments de réponses qui peuvent être avancés, les premiers sont hautement contextuels, à cette période peu anodine où le système libre a été conçu. En premier lieu, bien malin était celui qui aurait pu prédire les conséquences qu'auraient dans les années à venir la sortie de l'iPhone en novembre 2007 sur le marché de la mobilité numérique, par extension sur nos usages des objets techniques numériques aujourd'hui, *a fortiori* donc ceux des députés. Comment aurait-on pu anticiper, pour assurer aux postes des députés une compatibilité matérielle à toute épreuve, les cinq années suivantes d'innovations technologiques ? Que les arbitrages techniques aient à l'Assemblée nationale une durée de vie de cinq ans pose un premier problème, relatif au rythme rapide qui est celui de l'industrie des produits numériques, dont par ailleurs les députés sont friands. Également, il ne faut pas écarter le fait que 2007 fut la date d'entrée à grande échelle de l'informatique dans les bureaux parlementaires, et donc qu'auparavant les usages numériques des députés n'existaient qu'à la marge. En considérant que la grande majorité des députés-destinataires ne s'étaient pas encore confrontés à la polyvalence des fonctions possibles d'un système informatique,

1. On pourra certainement relier ceci au fait précédemment évoqué que Linagora a tardivement recruté son expert Kubuntu pour les ordinateurs de bureau, soit trois semaines après la signature du contrat avec l'Assemblée nationale.

2. Entretien téléphonique avec Michel-Marie Maudet, op. cit.

comment espérer si tôt savoir la place qui lui serait réservée au quotidien alors qu'eux-mêmes n'en avaient aucune idée ? Dès lors valait-il mieux choisir comme ici un outil aux usages prédéfinis et restreints pour accompagner les députés dans leurs nouveaux usages, ou proposer plutôt un outil multifonction qui aurait eu le mérite d'une plus grande adaptabilité à la variété des usages ?

Le choix du SSI pour la première solution reflète un positionnement propre à l'administration parlementaire qui, tout en veillant à ce qu'aucune condition matérielle n'entrave le travail des députés, parfois les chaperonne dans certains domaines complexes que ces derniers n'ont ni le temps ni le loisir d'appréhender par eux-mêmes. En ce sens le SSI a fourni un accompagnement resserré en proposant un outil parfaitement intégré à l'écosystème de l'Assemblée nationale, pour des usages orientés bureautique qui lui sont spécifiques. En somme c'est un outil de production d'information qui a été conçu dans cette version personnalisée de Kubuntu, en garantissant l'interopérabilité avec les formats de données en cours dans l'institution, en proposant un outil d'agenda spécifique, et en considérant comme secondaires les fonctionnalités multimédias et mobiles. Au contraire, les députés, une fois le stade de la découverte passé, ont montré davantage d'intérêt pour un outil de consommation d'informations, moins intégré à l'écosystème de l'institution qu'à celui global d'Internet. Ainsi, conformément à la nature intrinsèque de leurs fonctions, c'est de l'ensemble des informations circulant dans leur réseau d'influence qu'ils exigent de pouvoir prendre connaissance. Notons que cet état de fait ne change rien, sinon dans l'outil et le support, des pratiques parlementaires qui sont celles de ces élus depuis plusieurs décennies. Les 577 députés sont bien comme autant de chefs d'entreprises, qui certes ont leurs bureaux et certaines obligations au sein de l'institution parlementaire, mais dont l'essentiel de l'activité se fait en toute indépendance, et en premier lieu en circonscription.

Dès lors, hors de toute considération pouvant en éclairer, en expliquer ou en justifier les différents éléments, voici comment la situation s'est finalement présentée aux yeux du député une fois déployé le poste de travail. Alors qu'il leur aurait fallu un outil souple, ouvert et polyvalent, il leur a été fourni un système rigide, restreint et fermé. Rigide parce qu'incapable de s'adapter rapidement à l'évolution des technologies, restreint

parce que les usages effectifs ne pouvaient différer de ceux imaginés en amont, fermé parce que dédié à la communication interne à l'institution. Si l'on ajoute à cela ses résistances à être administré de façon centralisée, il apparaît que le système s'est montré finalement peu adapté aux différentes contraintes de la multiplicité des utilisateurs, alors qu'il paraissait initialement tout à fait adéquat à l'échelle de l'Assemblée nationale. Pour un utilisateur qui n'a aucune idée – et vraisemblablement peu d'intérêt pour la question – des restrictions inhérentes aux brevets et protections qui délimitent les frontières des usages numériques, et également des formats et normes avec lesquelles transitent les informations ou se connectent mutuellement les objets numériques, Ubuntu est apparu comme un système moins performant que Windows. Du point de vue des députés pour qui la connectivité sociale est la première qualité de l'élu, le logiciel libre, dont la connectivité technique défailait, manquait singulièrement d'attrait. En définitive, en partie en raison d'un contexte particulier, également du fait de certaines caractéristiques propres au logiciel libre, enfin aussi en raison de certains arbitrages effectués conjointement par le prestataire et le SSI, force est de constater que si un système GNU/Linux est synonyme de liberté du code, il n'a pas été dans ce cas synonyme de liberté des usages, *a fortiori* d'un monde de mobilité et d'interconnexions.

En conclusion

Le récit de la mise en œuvre du logiciel libre pour les postes des députés se termine ici quand, courant 2008, Linagora voit son contrat de maintenance reconduit, et ce successivement jusqu'à la fin de la treizième législature. D'autres et nombreuses péripéties pourraient encore être rapportées, si toutefois elles apportaient de nouveaux éléments à l'analyse, ce qui n'est pas le cas. Les autres enjeux, autres circonstances et autres arbitrages ne découlent que de ce qui a été à présent décrit, les longs mois qui ont suivi n'ayant en effet été consacrés qu'à finaliser l'ajustement entre les besoins des députés et les contraintes du nouveau système, puis enfin à veiller à la pérennité de l'ensemble.

D'un certain point de vue donc, le chantier est un succès, le défi technique de concevoir des postes de logiciels libres a été réalisé en dépit des contraintes de l'institution, qu'elles soient financières ou d'ordre

temporel¹. D'un autre côté, certains compromis n'ont jamais vraiment convaincu ni les députés ni le SSI, en particulier concernant les compatibilités matérielles avec les objets multimédia. Ce bilan mitigé du point de vue des usages explique comment en juin 2012, à l'aube de la législature suivante, le Collège de la Questure annonce sa décision de ne pas renouveler telle quelle l'expérience, pour laisser cette fois-ci aux députés le choix du système d'exploitation qu'accueilleront leurs futurs postes. Ce dernier élément entérine le fait que, politiquement, au regard du projet initial, Bernard Carayon et Richard Cazenave considèrent ce projet comme un échec².

Si le logiciel libre a permis qu'un poste de travail générique devienne un poste de travail profondément spécifique, c'est moins pour répondre à la demande de ses destinataires qu'à celle de son commanditaire. Ce qui explique à terme le succès très relatif de ce projet tient alors en ce que la souplesse du logiciel libre ne fut surtout exploitée que dans le sens des impératifs d'une administration, au détriment de ceux des usagers. Il est frappant dans ce récit de constater que le libre est pris dans une vaste problématique où les acteurs en présence se renvoient mutuellement le reflet de leur rigidité respective. Alors que l'institution s'emploie à modeler un outil autour de la rigidité perçue de l'utilisateur, ce dernier proteste de la rigidité des usages qui y sont prescrits, et les concepteurs de regretter la rigidité aussi bien de l'institution, son commanditaire, que des députés, ses destinataires. Face à ces contraintes fortes, la souplesse du logiciel libre fut largement éprouvée, et par extension celle de Linagora qui en avait la maîtrise. Le logiciel libre est alors la variable d'ajustement qui a permis à ces différents acteurs de s'entendre autour de ce projet.

En définitive, la liberté du code a été ici employée à figer l'évolution des pratiques techniques, dans une forme considérée comme sûre, fiable et non problématique. Que l'expérience utilisateur qui a servi ici de référence – celle de Windows 95 – soit une interface conçue pour des usages datant de près de douze ans – soit une période très longue dans l'histoire de l'informatique – est symptomatique d'une inertie forte de l'administration parlementaire. En ne concédant que des évolutions minimales, et seulement à chaque nouvelle législature, de son mode d'action et de

1. Entretien téléphonique avec Michel-Marie Maudet, *op. cit.*

2. Entretien téléphonique avec Bernard Carayon, *op. cit.*

son fonctionnement, l'institution observe les effets de mode sans y souscrire, et fait ainsi preuve d'une constance assurément gage de pérennité. Néanmoins, cette logique qui a pourtant fait ses preuves se trouve mise à mal par l'accélération du rythme des innovations technologiques ces dernières années, qui creuse sérieusement l'écart entre les usages techniques en cours hors de l'institution et les usages internes. Voilà comment, alors que les députés revendiquent leur ancrage dans la société de l'information, la liberté des usages technologiques s'est trouvée compromise par cette mise en œuvre du logiciel libre, le faisant paradoxalement apparaître à leurs yeux comme coercitif, et même anachronique.

Remerciements

Que soient ici remerciés l'ensemble des interlocuteurs – cités dans le texte ou pour la plupart d'entre eux restés (à leur demande) anonymes – qui ont accepté de me confier leur témoignage. Également, il me faut saluer la confiance accordée par quelques-unes de mes rencontres, dont le crédit accordé à mes recherches fut déterminant pour lever l'opacité entourant cette enquête.

Références

- ABÉLÈS, Marc, « Pour une anthropologie des institutions », dans : *L'Homme* 35.135 (1995), p. 65–85.
- *Un ethnologue à l'Assemblée*, Paris : Odile Jacob, 2001.
- AKRICH, Madeleine, « Comment décrire les objets techniques ? », dans : *Techniques & Culture. Revue semestrielle d'anthropologie des techniques* 54-55 (1987), p. 205–219, URL : <http://tc.revues.org/4999>.
- CARAYON, Bernard, *Rapport d'information sur la stratégie de sécurité économique nationale*, rap. tech. 1664, Paris : Assemblée Nationale, 2004.
- *À armes égales. Rapport au Premier ministre*, Paris : La Documentation Française, 2006.

- CARON, André H., Luc GIROUX et Sylvie DOUZOU, « Diffusion et adoption des nouvelles technologies : le micro-ordinateur domestique », dans : *Canadian Journal of Communication* 11.4 (1985), p. 369–389, URL : <http://www.cjc-online.ca/index.php/journal/article/view/403>.
- COURTY, Guillaume, éd., *Le travail de collaboration avec les élus*, Paris : Michel Houdiard Editeur, 2005.
- DENOUEËL, Julie et Fabien GRANJON, *Communiquer à l'ère numérique. Regards croisés sur la sociologie des usages*, Paris : Presses des Mines, 2011.
- FRAYSSINET, Jean, *La bureaucratie. L'administration française face à l'informatique*. Paris : Berger-Levrault, 1981.
- GUILLEMIN, Christophe, « Linux : Ubuntu s'installe sur les PC des députés », dans : *ZDNet* (2007), URL : <http://www.zdnet.fr/actualites/linux-ubuntu-s-installe-sur-les-pc-des-deputes-39367717.htm>.
- JAMOUS, Haroun et Pierre GRÉMION, *L'ordinateur au pouvoir. Essai sur les projets de rationalisation du gouvernement et des hommes*, Paris : Seuil, 1978.
- LATOUR, Bruno, *La science en action : Introduction à la sociologie des sciences*, Paris : Editions La Découverte, 2005.
- MILAN, Anthony, « Migration de la gendarmerie sous OOo », dans : *LinuxFr* (2005), URL : <http://linuxfr.org/news/migration-de-la-gendarmerie-sous-ooo>.
- ROCARD, Michel et Bernard CARAYON, « L'encodage des œuvres numériques, un nouveau Big Brother ? », dans : *Le Figaro* (juin 2006).
- URVOAS, Jean-Jacques et Alexandre MAGALI, *Manuel de survie à l'Assemblée nationale*, Paris : Odile Jacob, 2012.
- ZETLAOUI, Tiphaine, *Le Parlement face aux TIC*, Quaderni, t. 75, Paris : Maison des sciences de l'homme Paris, 2011.

RTFM ! La chimie computationnelle : une communauté scientifique loin du libre

Alexandre HOCQUET

1. RTFM ! What FM ?

Read The Fucking Manual ! Cet acronyme geek est ce que l'on répond, excédé, lorsqu'une personne pose une question sans s'être donné la peine de chercher. Il est particulièrement courant sur les forums ou les listes de diffusion dans lesquelles on parle de logiciel. Mais le manuel (ou son absence) représente aussi autre chose. Aux débuts de l'histoire du logiciel, dans les années cinquante et soixante, une grande partie du développement était réalisée dans des laboratoires universitaires et des entreprises par des scientifiques et des ingénieurs. Il était normal à l'époque que le logiciel fasse partie intégrante de l'ordinateur pour lequel il avait été conçu. La portabilité des applications entre deux machines était quelque chose de très rare, et l'idée d'échanger, de vendre ou encore l'octroi de licences de logiciels ne venait jamais à l'esprit des scientifiques et ingénieurs, universitaires ou industriels, qui les concevaient. Ce n'est qu'à partir des années soixante que le logiciel est devenu un produit qui pouvait être acheté séparément de l'ordinateur lui-même. En 1967, l'expression *software crisis*

(crise du logiciel) est apparue pour la première fois dans le milieu des développeurs pour caractériser les difficultés rencontrées par l'industrie du logiciel. De manière complètement opposée à l'évolution spectaculaire des performances et du coût du matériel, les développements de logiciels étaient de plus en plus longs, nécessitaient de plus en plus de main d'œuvre, coûtaient de plus en plus cher. Les erreurs (les bogues/bugs) étaient de plus en plus difficiles à localiser et à corriger étant donnée la complexité croissante du code et des langages, et les révisions des versions avaient de plus en plus de difficultés à suivre l'évolution du matériel. La situation de crise ne s'est en fait jamais atténuée et la *software crisis* est, depuis lors, devenue un art de vivre pour les développeurs plutôt qu'un épisode passager¹.

Avec la popularisation de l'ordinateur, il est arrivé un moment où les logiciels ont dû être utilisés par des personnes différentes de ceux qui les concevaient. La question de la documentation, ou plus généralement de l'adéquation entre les pratiques des utilisateurs et les pratiques des développeurs (le *fucking manual*) est aussi un emblème des problèmes jamais résolus de la crise du logiciel : il s'est donc avéré que les plus grandes tensions dans l'industrie informatique (mais aussi dans l'histoire du calcul informatique en tant qu'activité de recherche scientifique) étaient liées à la question du logiciel. Depuis la crise du logiciel des années soixante, la maintenance, ou le service après vente, ou encore la gestion de la communauté, selon le point de vue duquel on se place, est un des points critiques dans les difficultés de production et de distribution de logiciels ainsi qu'une des sources de tensions entre développeurs et utilisateurs. Le manuel, ou, plus largement, la question de savoir comment utiliser un logiciel écrit par quelqu'un d'autre, est une de ces tensions. RTFM !, c'est aussi l'illustration de cette tension.

Dans le cadre de la recherche scientifique, ces tensions existent aussi et sont devenues particulièrement importantes quand l'ordinateur est devenu, dans le laboratoire, un objet quotidien, un objet de masse. Les idées et les pratiques du logiciel libre sont nées de ces tensions comme le suggère l'anecdote de Richard Stallman et ses déboires avec les pilotes d'imprimante Xerox. Mais ces tensions ont aussi parfois conduit les cher-

1. Nathan L. ENSMENGER, *The Computer Boys Take Over : Computers, Programmers, and the Politics of Technical Expertise*, Cambridge : MIT Press, 2010.

cheurs vers des voies complètement différentes du libre, montrant ainsi que monde académique et développement du libre ne font pas forcément bon ménage. C'est le cas de la chimie computationnelle¹.

2 La chimie computationnelle

Qu'est-ce que la chimie computationnelle ? Il s'agit du domaine scientifique défini comme l'utilisation des ordinateurs pour résoudre les problèmes de chimie. Si on cherche les mots *computational chemistry* dans Google Images, on obtient des images colorées de molécules ou de propriétés moléculaires littéralement en train de sortir hors de l'écran. C'est qu'il s'agit d'un domaine de la recherche scientifique pour lequel l'esthétique des représentations de molécules par les terminaux graphiques est souvent utilisé pour donner une image moderne et hautement technologique de la recherche.

D'une part, la chimie computationnelle remonte à l'équation de Schrödinger². La « chimie théorique » ou « chimie quantique » consistait dans les années trente et quarante à appliquer le formalisme de la mécanique quantique pour comprendre la structure de systèmes simplissimes (comme la molécule d'hydrogène), en résolvant des équations sur un tableau noir ou une feuille de papier. Avec l'avènement des premiers ordinateurs, les calculs de chimie quantique ont pu être effectués sur les rares heures de calcul disponibles des superordinateurs. À l'époque, il s'agissait d'objets rares et chers, emblèmes de la *big science*, financés essentiellement par des crédits fédéraux ou nationaux. Sur ces superordinateurs, le temps de calcul disponible était partagé entre les différents utilisateurs. Les chimistes théoriciens n'étaient alors que des acteurs mineurs dans ce do-

1. « Chimie computationnelle » est un néologisme, traduction littérale de *computational chemistry*. Il est employé au Canada mais peu en France où l'on trouve parfois les termes « chimie informatique » ou « chimie numérique ». Mais les mots informatique ou numérique sont aussi liés à des époques. L'adjectif *computational* sied parfaitement aux années quatre-vingt et quatre-vingt-dix et au monde anglo-saxon, époque et lieu où se déroule notre histoire.

2. L'équation de Schrödinger est l'équation de base de la mécanique quantique. Datant de 1927, elle représente ce que doit résoudre un calcul dans le cadre d'un formalisme mathématique quantique, infiniment plus complexe pour le traitement par l'ordinateur que le formalisme mathématique de la physique classique.

maine, les calculs de chimie étant considérés comme moins stratégiques que la balistique ou l'aérodynamique, par exemple.

Par ailleurs, dans les années soixante et soixante-dix, une conception complètement différente de la façon de modéliser les molécules a émergé. Cette vision ressemblait à celle des modèles moléculaires tridimensionnels que les chimistes manipulent en classe ou au laboratoire et était basée sur un modèle énergétique de potentiel harmonique similaire à des ressorts. Elle a été nommée « mécanique moléculaire », en référence à une vision newtonienne des mouvements des molécules. Il n'y avait ni électrons ni réactivité dans ce modèle, mais la possibilité d'exécuter des calculs sur des molécules plus grosses (et donc plus utiles, comme par exemple les protéines) explique son succès immédiat quand les premiers terminaux graphiques ont permis aux scientifiques d'afficher effectivement des molécules sur l'écran dans les années soixante-dix.

Les modèles mathématiques de la chimie quantique, basée sur la mécanique quantique, comportent quant à eux des procédures mathématiques trop compliquées pour être calculées à la main, y compris pour les molécules les plus simples. L'apparition des premiers ordinateurs a laissé entrevoir l'espoir de résoudre l'équation de Schrödinger avec des méthodes numériques, de manière approchée. Mais il est vite devenu clair que la demande en puissance de calcul (exponentielle en fonction du nombre d'atomes des molécules) condamnait la chimie quantique à ne s'intéresser qu'aux petites molécules, malgré les perspectives inouïes de développement des performances du matériel informatique. La mécanique moléculaire, au contraire, s'appuie sur la physique newtonienne simple, permettant des calculs numériques sur les molécules les plus intéressantes, et s'appuie sur une façon de définir et de représenter les molécules familière aux chimistes. Le modèle ne repose pas sur des bases théoriques solides (contrairement à la chimie quantique). Il est, au contraire, une définition simple des liaisons et des atomes¹. Il s'agit donc de deux concepts

1. En mécanique moléculaire, les liaisons sont considérées énergétiquement comme des potentiels harmoniques (c'est-à-dire des boules et des ressorts, inspirés par l'idée de mouvements moléculaires internes de la spectroscopie infra rouge). La mécanique moléculaire est aussi une autre façon de définir ce qu'est une molécule : Contrairement à la description de la chimie quantique pour laquelle une molécule est une collection de particules, négativement (électrons) ou positivement (noyaux) chargées, la mécanique moléculaire correspond à la vision classique des chimistes pour décrire les molécules (atomes liés par des liaisons) et

complètement distincts de modélisation, deux théories différentes. Avec l'avènement du terminal graphique, la possibilité de voir la structure tridimensionnelle des molécules sur un écran explique le succès de la mécanique moléculaire. Il a été immédiat, non seulement pour la chimie en tant que science computationnelle (à partir du moment où l'ordinateur a été accessible aux chimistes), mais aussi pour les constructeurs de matériel qui voyaient dans cette activité de recherche une publicité moderne et scientifique.

En ce qui concerne le matériel, la chimie computationnelle est donc un domaine scientifique qui a commencé comme un client mineur des ressources de calcul des superordinateurs des années soixante, mais s'est rapidement développé pour devenir un acteur dominant dans le domaine du calcul scientifique lorsque les terminaux graphiques sont devenus disponibles vingt ans plus tard. Lorsque l'ordinateur personnel (le PC) et les stations de travail ont popularisé l'informatique dans les laboratoires de recherche, une nouvelle ère de la « modélisation de bureau » (*desktop modeling*) a coïncidé, pour les chimistes computationnels, avec une demande croissante de logiciels de modélisation de la part de l'industrie pharmaceutique, et un investissement dans ce marché potentiel de la part des fabricants de matériel, y compris des géants comme IBM.

Par ailleurs, pendant les années quatre-vingt, les années Reagan, les universités américaines, bon gré mal gré, lançaient des projets de « transfert de technologie » (*technology transfer*). Les logiciels produits dans les universités commençaient à être considérés comme des revenus potentiels par ces dernières, à travers leur politique de brevet ou de licences exclusives portant sur les résultats de leurs activités de recherche. Le logiciel de modélisation moléculaire¹ est devenu l'un des ces objets susceptibles de rapporter de l'argent et ainsi est devenu un objet central dans la communauté des chimistes computationnels. Pour la première fois, les scienti-

inventer un modèle mathématique simple compatible avec cette définition d'une molécule. La complexité de la mécanique moléculaire réside plutôt dans sa difficulté à paramétrer des dizaines de types d'atomes, des centaines de liaisons, et des milliers de paramètres créés pour ajuster le modèle mathématique simple aux résultats expérimentaux de la structure moléculaire. Son inconvénient est l'absence totale d'électrons dans la définition de la molécule et donc de la simple idée de réactivité (ce qui est un gros manque pour un chimiste)

1. Le logiciel de « modélisation moléculaire » est le logiciel conçu par et pour les chimistes computationnels. Il englobe un modèle (de chimie quantique et/ou de mécanique moléculaire) et permet de calculer la structure et/ou les propriétés de molécules.

fiques qui concevaient des logiciels de modélisation moléculaire n'étaient plus les mêmes que ceux qui accomplissaient les calculs. Deux catégories distinctes de scientifiques concernés par la chimie computationnelle se formaient : les développeurs d'un côté, les utilisateurs de l'autre. La question « comment transférer le logiciel de modélisation moléculaire du développeur vers l'utilisateur » est ainsi née... et a suscité beaucoup de tensions dans la communauté.

3 Le terminal graphique

L'impact matériel décisif sur cette redéfinition des rôles a été l'avènement du terminal graphique dans les années soixante-dix. La capacité de voir une molécule en trois dimensions sur un écran au lieu de coordonnées sur une feuille de papier d'imprimante s'est avérée essentielle à la réussite de la chimie computationnelle en tant que discipline scientifique (et en particulier des modèles de mécanique moléculaire qui pouvaient représenter de plus grandes et plus belles molécules). En tant que discipline scientifique, mais aussi en tant que promoteur de ce nouveau matériel : les belles images de molécules sur un écran représentaient à l'époque la publicité la plus esthétique pour une utilisation scientifique des écrans graphiques de haute résolution. D'un côté, la chimie computationnelle suscitait l'intérêt des décisionnaires de recherche et développement de l'industrie pharmaceutique, qui voyaient dans la modélisation la « promesse technologique » qui allait faire baisser les coûts énormes de découverte et de mise sur le marché de nouveaux médicaments. De l'autre, les fabricants de matériel voyaient dans ce nouveau domaine scientifique un allié pour vendre la nouvelle technologie.

Au début des années quatre-vingt, la chimie computationnelle n'était plus un acteur mineur des ressources informatiques. Le tiers des heures des centres de calcul américains était consacré aux calculs en chimie computationnelle. Les chimistes computationnels étaient devenus importants, et étaient ainsi concernés par le monde commercial et industriel, et ce d'une triple façon.

D'abord, les chimistes computationnels étaient impliqués dans l'industrie de la fabrication d'ordinateurs. Les géants comme IBM ont vu dans

la chimie computationnelle une force majeure dans le domaine du calcul intensif, et IBM a activement financé les heures de calcul de groupes de recherche en chimie, mais aussi a investi une partie substantielle de ses efforts de R&D à l'élaboration d' *interactive molecular graphics*¹. Dans le même ordre d'idée, IBM a également été un important investisseur dans l'un des grands éditeurs de logiciels de modélisation moléculaire, Polygen. Mais les chimistes computationnels ont également été parmi les meilleurs clients (et les meilleurs publicitaires) pour les fabricants de stations de travail aux puissantes capacités graphiques tels que Silicon Graphics et bien d'autres.

Ensuite, un marché potentiel pour la modélisation par la mécanique moléculaire a été pressenti par les chimistes du secteur pharmaceutique, ce qui a impliqué la création de quelques entreprises de logiciels à fortes ressources financières dans les années 1980. L'intérêt croissant de l'industrie pharmaceutique pour cette « promesse technologique » a mobilisé les domaines de la chimie déjà très liés à la pharmacie. En écho, les différents choix stratégiques des développeurs de logiciels en matière de politique de brevet ou de licences exclusives ont été influencés par les pratiques traditionnelles de l'industrie pharmaceutique, connue pour être l'une des plus secrètes et compétitives.

Enfin, ils étaient concernés par le logiciel vu comme une entreprise commerciale. Dans les années 1980 et 1990, les *success stories* des entreprises du domaine de l'ordinateur personnel (comme Microsoft et Autodesk) étaient autant d'exemples incitant les développeurs mais aussi et surtout les administrations des universités à se lancer dans l'exploitation commerciale de leurs logiciels.

Dans le même temps, les informaticiens qui ont imaginé les bases du logiciel libre et d'un Internet neutre, dans le monde académique ou non, représentaient une communauté scientifique de pairs, de coopération entre spécialistes, partageant des valeurs communes et séparée du reste de la société. Au tournant des années soixante-dix et quatre-vingt, cette communauté était institutionnellement, technologiquement, géographiquement proche des endroits où s'est construite simultanément la communauté

1. Eric FRANCOEUR et Jérôme SEGAL, « From model kits to interactive computer graphics », dans : *Models : The Third Dimension of Science*, sous la dir. de Soraya de CHADAREVIAN et Nick HOPWOOD, Palo Alto, 2004.

des chimistes computationnels. Pourtant, cette « république des informaticiens », décrite par le sociologue Patrice Flichy, était immergée dans un contexte économique et politique complètement différent, et possédait un « imaginaire » tout aussi différent de celui en vigueur en chimie¹.

4. Cinq pratiques caractéristiques du Libre

Alors que les développeurs de logiciels en chimie computationnelle cherchaient différentes voies pour distribuer leurs produits, la communauté scientifique à laquelle ils appartenaient n'avait que peu de relations (professionnelles ou militantes) avec les acteurs du mouvement du logiciel libre naissant. Le travail de l'anthropologue Christopher Kelty a été de s'intéresser à la question de savoir « comment » le mouvement/phénomène du logiciel libre marche, plutôt que savoir « pourquoi ». C'est-à-dire qu'il s'est intéressé aux pratiques des gens impliqués dans le logiciel libre, plutôt que de tenter d'expliquer son (éventuel) succès. En référence au concept de *public sphere* d'Habermas (qu'on peut traduire par espace public), Kelty propose l'idée de public récursif (*recursive public*) pour définir ce que sont ces pratiques caractéristiques². Récursif, parce que ce public n'a pas seulement un discours sur la technologie : il utilise cette technologie pour produire son action politique, ou son activité en général, et son discours même. Récursif, parce que cette technologie dépend de différentes couches successives imbriquées, toutes concernées d'une manière ou d'une autre par le libre : logiciels d'application, compilateurs, langages, systèmes d'exploitation, matériel, protocoles Internet. . .

Kelty affirme qu'il y a cinq pratiques fondamentales qui définissent le public récursif du logiciel libre. Il s'agit bien de pratiques (et non de règles ou de normes), puisque d'une part, le travail de Kelty a consisté à observer en ethnologue les acteurs du logiciel libre dans leur milieu, et d'autre part, parce que Kelty estime plus pertinent, pour décrire les « gens du libre »,

1. Patrice FLICHY, « Internet ou la communauté scientifique idéale », dans : *Réseaux* 17.97 (1999), p. 77–120.

2. Christopher KELTY, *Two bits : the cultural significance of free software*, Durham : Duke University Press, 2008.

de recourir à ce qu'ils ont en commun dans ce qu'ils font plutôt que dans ce qu'ils disent ¹.

Première pratique : partager le code source. La première pratique, inscrite comme principe de base par les acteurs du libre eux-mêmes, consiste à développer une communauté en partageant, donc en rendant ouvert le code source. Le fonctionnement du logiciel est ainsi accessible à tous.

Deuxième pratique : conceptualiser des systèmes ouverts. Pour construire un public récuratif, le maximum d'objets technologiques doivent être ouverts, de l'ordinateur à l'Internet en passant par toutes les couches logicielles. Toutes ces technologies appartenant à des domaines différents sont rendues ouvertes chacune à leur façon (TCP-IP, WWW,...), c'est-à-dire qu'il faut inventer une idée de ce qu'est « être ouvert » dans des domaines différents, pour que la communauté puisse évoluer dans un milieu adapté au public récuratif.

Troisième pratique : écrire des licences. Pour chaque système ouvert conceptualisé, la façon de le partager et de l'inclure dans une des couches technologiques est d'en écrire la licence. Écrire des licences matérialise les conditions selon lesquelles les objets ouverts circuleront dans la communauté.

Quatrième pratique : coordonner des collaborations. Comme le bazar décrit par Eric S. Raymond, les communautés autour du logiciel libre ne sont pas caractérisées par leur organisation hiérarchique ². Pour autant, mettre en œuvre des pratiques collaboratives fait partie de l'essentiel des mouvements du Libre. Les projets libres ne peuvent être viables que si la collaboration entre les acteurs est coordonnée. Cela n'implique pas forcément des organisations, mais pour le moins de l'adaptabilité.

1. Cette description des pratiques définies par Kelty est inspirée de celle faite par Andreas Lloyd qu'on peut trouver sur son blog : <http://andreaslloyd.dk/2008/11/bit-by-bit>. Par ailleurs, un autre chapitre du présent livre traduit un des chapitres du livre de Kelty (*Two Bits*).

2. Eric S. RAYMOND, *The Cathedral and the Bazaar*, Sebastopol, CA : O'Reilly, 2001, URL : <http://www.catb.org/esr/writings/homesteading/>.

Cinquième pratique : fonder des mouvements. Les communautés du libre partagent aussi en commun la motivation de créer à partir de leur projet une façon de voir le libre. Construire une narration, voire être prosélyte, sont aussi des pratiques caractéristiques du libre.

La communauté des chimistes computationnels a quelque chose d'un public récuratif : c'est une communauté dont l'objet technologique (le logiciel) est imbriqué au cœur même de son existence dans plusieurs couches technologiques successives, essentielles à la définition de l'activité de la communauté. Le logiciel de modélisation moléculaire est d'abord lié à l'ordinateur et la transformation de ce dernier d'objet rare et réservé à des experts en un objet accessible et modulable est essentielle. Le logiciel est particulièrement lié au terminal graphique, non seulement en tant qu'objet technologique définissant les caractéristiques du logiciel, mais aussi en tant qu'objet technologique façonnant la communauté elle-même par ses implications économiques et commerciales.

Pour autant, la communauté des chimistes computationnels, par opposition à la « république des informaticiens » est une communauté pour laquelle les cinq pratiques de Kely ont du mal à faire l'unanimité. On peut même avancer que l'ensemble des cinq pratiques caractéristiques, qui soude la communauté du libre, divise la communauté des chimistes computationnels. Qu'il s'agisse de partager le code source, de conceptualiser l'ouverture des systèmes, d'écrire des licences, de coordonner des collaborations ou de fonder des mouvements, aucun consensus ne s'est dégagé et qu'au contraire, chacun de ces cinq sujets est source de tensions.

5. Trois contraintes à la pérennité du Libre

Dans une étude sociologique sur les logiciels économétriques, Gambar-della et Hall avancent que « le logiciel (scientifique) librement partagé » peine à s'imposer comme pratique de distribution dans certaines conditions bien précises : lorsque les attentes de profits pécuniaires sont élevées, lorsque que la demande du marché potentiel d'utilisateurs est importante,

et quand les normes incitant au partage sont faibles ou difficiles à appliquer¹.

Pour que le partage libre de logiciels soit un succès, il faut qu'existe une sorte de coordination (comme la Free Software Foundation), ou ce que Kelty appelle « fomenteur un mouvement » dans son approche anthropologique. Au contraire, dans la communauté de la chimie computationnelle, la variété des licences imaginées par les différents développeurs (ou leurs administrations), le faible nombre de références à la FSF ou le faible nombre de recours à ses licences comme la GPL exprime les tensions et l'absence de consensus.

Un profit potentiel élevé et une demande forte du marché sont les deux autres ingrédients de la recette pour faire obstacle au « libre partage de logiciels ». Les liens étroits entre la chimie computationnelle et l'industrie des constructeurs d'ordinateurs, d'une part, et avec l'industrie pharmaceutique, d'autre part, correspondent tout à fait à ces deux critères : d'un côté les constructeurs ont vu dans le logiciel de modélisation moléculaire un objet pouvant rapporter gros en ventes de matériel (particulièrement pour populariser les terminaux graphiques). Accessoirement, le logiciel de modélisation moléculaire pouvait rapporter gros aux développeurs de logiciels eux-mêmes. D'un autre côté, dans l'industrie pharmaceutique, la demande du marché était importante. L'industrie pharmaceutique était prête à payer cher, appartenait à une culture de recherche et développement très versée dans le secret ou dans les brevets (dans le but de protéger leurs découvertes), et donc avait une idée *a priori* du logiciel en tant qu'objet scientifique proche de cette culture². Les incitations pour les développeurs à embrasser l'esprit d'entreprise, mais aussi d'avoir recours à des licences fermées, étaient ainsi élevées. À cela, on peut ajouter la tendance des universités américaines, à travers leurs administrations de « transfert de technologie » ou leurs « bureaux des brevets » à pousser vers la commercialisation. Ces incitations se produisaient parfois en harmonie avec les projets des équipes de recherche qui développaient des logiciels,

1. Alfonso GAMBARDELLA et Bronwyn H. HALL, « Proprietary versus public domain licensing of software and research products », dans : *Research Policy* 35.6 (2006), p. 875–892.

2. D. C. MOWERY et al., « The growth of patenting and licensing by US universities : an assessment of the effects of the Bayh-Dole act of 1980 », dans : *Research policy* 30.1 (2001), p. 99–119.

parfois contre leur gré, que ce soit contre la volonté de certains universitaires de faire du libre, ou en leur imposant des choix de licences pas forcément pertinents.

Le contexte politique, académique et économique de la communauté, mis en lumière par l'étude de Gambardella et Hall éclaire pourquoi il a été si difficile pour les chimistes computationnels de « fomenteur des mouvements » au sens de Kelty : les normes du libre, telles que celles de la FSF, ont besoin d'un contexte académique adéquat pour pouvoir se répandre et s'imposer dans la communauté scientifique. Mais le logiciel et sa conceptualisation, et donc les pratiques de la communauté, sont aussi intimement liés à l'ordinateur lui-même, au matériel.

6. La rupture du *desktop modeling*

À l'époque où l'informatique consistait à attendre du temps de calcul sur un super-ordinateur, « la manipulation » ou l'exploration des modèles (*model tweaking*), c'est-à-dire l'exécution itérative de versions de modèles modifiées de manière mineure à chaque fois, était impraticable. Le super-ordinateur n'était accessible qu'à travers de nombreux intermédiaires.

L'époque suivante¹, selon les catégories de Lenhard et Johnson² est l'époque de la disponibilité des machines. Des ordinateurs de plus en plus petits avec des puissances de calcul de plus en plus grandes, envahissaient non seulement le laboratoire, mais aussi envahissaient le bureau même des scientifiques, y compris celui de ceux qui n'étaient pas spécialistes en informatique. L'informatique de bureau, telle que popularisée par l'ordinateur personnel d'IBM (le *PC*) et plus encore par les stations de travail (*workstations*) avec leurs meilleures capacités graphiques, pouvaient finalement transformer le travail des chimistes computationnels. Ils et elles leur donnaient le pouvoir de faire leurs propres calculs à portée de leurs claviers. Ils pouvaient enfin explorer les modèles, puisque les modèles

1. L'article de Johnson et Lenhard fait partie d'un livre sur le thème d'un changement d'époque (*epochal break*) de l'activité scientifique.

2. Ann JOHNSON et Johannes LENHARD, « Toward a new culture of prediction : Computational modeling in the era of desktop computing », dans : *Science Transformed ? Debating Claims of an Epochal Break*, sous la dir. d'Alfred NORDMANN, Hans RADDER et Gregor SCHIEMANN, Pittsburgh : University of Pittsburgh Press, 2011, p. 189–200.

leur étaient enfin accessibles sans intermédiaire, dans leur propre ordinateur personnel.

La nature exploratoire de la modélisation est donc tributaire de l'accès à bas prix, pratique et local à la puissance informatique. Cette accessibilité n'est venue qu'avec la maturité de l'ordinateur de bureau (*desktop computer*), en particulier au moment où ces ordinateurs ont été mis en réseau, avec l'avènement d'Internet. Historiquement, cela signifie que le changement dans la pratique de la modélisation s'est produit plutôt dans les années quatre-vingt-dix (l'époque de la popularisation de l'ordinateur personnel) que dans les années cinquante (l'époque des super-ordinateurs, de l'introduction de l'informatique dans la pratique scientifique), ou que dans les années soixante-dix (l'époque de l'utilisation généralisée des ordinateurs centraux (*mainframe*) en recherche). Jusqu'aux années quatre-vingt, la modélisation était un type de calcul qui était commun à toutes sortes de domaines, et constituait une branche indépendante de la recherche (*computing*). À partir des années quatre-vingt, la modélisation dans les sciences a été intégrée entre théorie et expérience et s'est imposée comme une dimension indispensable de l'entreprise scientifique, spécifique à chaque domaine (*computation*). Avec l'omniprésence de l'ordinateur de bureau, l'exploration des modèles est devenue une pratique essentielle des ingénieurs ou des chercheurs utilisant l'ordinateur (*computational scientists*), et pas seulement un domaine réservé aux professionnels de l'informatique (*computing scientists*). Le *desktop modeling* est leur façon de modéliser. Ce mode d'exploration dépend aussi de la capacité des chercheurs à évaluer rapidement les résultats de différentes versions de modèles. L'évaluation des résultats du modèle est particulièrement possible depuis l'apparition du terminal graphique, comme dit plus haut, grâce à la visibilité des résultats de structures de molécules, par opposition aux résultats sur imprimante. La facilité d'accès pour l'utilisateur (*user-friendliness*) ainsi créée favorise le bricolage du modèle par l'utilisateur.

Il y a deux conséquences à cela du point de vue de la relation entre le monde du libre et les chimistes computationnels. La première est qu'on voit apparaître la frontière entre la « république des informaticiens » décrite plus haut, qui correspond aux *computing scientists* et les chimistes computationnels qui font partie des *computational scientists*. Cette fron-

tière correspond donc aussi à un rapport différent à l'ordinateur, d'abord dans la relation avec le matériel, ensuite dans la relation avec le logiciel.

Dans le second cas, Lenhard et Johnson parlent de « marchandisation » (*commodification*) de logiciels scientifiques. Ils se réfèrent à une étude antérieure de l'historien des sciences Terry Shinn qui introduit le concept de « technologie-recherche » (*research-technology*) pour les objets scientifiques à la frontière du monde de la recherche et du monde industriel. Shinn conceptualise la « transversalité », pour définir des instruments d'un genre spécifique, comme par exemple l'ultra-centrifugeuse¹, et Lenhard et Johnson font valoir que le logiciel de modélisation de l'époque du *desktop modeling* correspond tout à fait aux caractéristiques définies par Shinn et il est tentant d'appliquer l'idée aux logiciels de modélisation de la chimie computationnelle :

- Il est conçu dans des communautés interstitielles : à l'interstice des informaticiens et des chimistes, à l'interstice entre le monde académique et le monde industriel, comme l'ultra-centrifugeuse a été conçue au cours d'une thèse académique sur la physique quantique puis a servi dans le domaine de la biologie, académique et industriel.
- c'est un dispositif générique : il peut servir à plusieurs usages. Le logiciel de modélisation sert dans la recherche fondamentale pour développer les modèles eux mêmes ou pour étudier les propriétés de molécules ou de matériaux, organiques, inorganiques, biomoléculaires ou macromoléculaires. Il sert aussi dans l'industrie, pour la rationalisation du design de médicaments, comme l'ultracentrifugeuse sert à la biologie, la séparation isotopique...
- il est utilisé de manière isolée de son contexte d'invention : dans le cas du logiciel, c'est exactement la tension provoquée par la différence entre le développeur et l'utilisateur.
- il est porteur de nouvelles métrologies. Dans le cas de l'ultracentrifugeuse, l'appareil de « recherche-technologie », a imposé son vocabulaire scientifique aux domaines très différents les uns des autres dans lesquels il a essaimé. Le *desktop modeling* comme le définissent Lenhard et Johnson n'est pas seulement la vulgarisation et la démocratisation de l'outil informatique. Cela correspond aussi à un changement

1. Bernward JOERGES et Terry SHINN, *Instrumentation Between Science, State and Industry*, Boston : Kluwer Academic Publishers, 2001.

radical dans la manière de mener des études de modélisation, et le logiciel de modélisation incarne l'objet de « recherche-technologie » qui porte les métrologies nouvelles : une nouvelle façon de modéliser.

Cette nouvelle façon de modéliser admet que la structure et la théorie du modèle ne déterminent pas complètement le calcul de modélisation. De nombreux paramètres ou procédures de calcul ne sont pas figés, ni déterminés par les seules théories mathématiques qui sous-tendent le modèle. On a vu plus haut qu'en mécanique moléculaire, la quantité de paramètres à définir puis évaluer puis calibrer est énorme. Même en chimie quantique, les procédures numériques destinées à résoudre les équations sont difficilement mathématisables. Selon les attentes et les besoins des utilisateurs, selon les contingences de la programmation ou du matériel, selon la disponibilité ou la publication de paramètres ou de procédures essentiels, les caractéristiques du modèle dans le logiciel peuvent être opaques et avoir besoin de tests, de comparaisons, d'adaptations, en un mot de manipulations avant de pouvoir être opérationnellement utilisées. L'appellation de « boîte noire » pour désigner cette opacité n'est pas juste le regret de ne pouvoir en maîtriser tous les paramètres pour l'utilisateur. C'est aussi l'essence même de la manipulation des modèles.

L'idée de logiciel « boîte noire » est souvent indiquée comme à déplorer du point de vue de la reproductibilité scientifique. Ne pas connaître les procédures exactes à l'intérieur de la « boîte noire » empêche le scientifique de contrôler ses calculs donc sa propre production scientifique. Les notions d'*Open Science* ou le *Science Code Manifesto* sont des initiatives récentes pour adapter les idées du libre à ce que devrait être l'activité scientifique, bien au-delà de l'utilisation de logiciels libres ou propriétaires. Il s'agit ici de la deuxième pratique conceptualisée par Kelty : conceptualiser des systèmes ouverts. Les initiatives comme l'*Open Science* sont des réflexions sur les pratiques de l'activité scientifique définies par la communauté elle-même, la communauté scientifique en tant qu'espace public.

Ce « peaufinage des modèles » n'est donc possible que si l'ordinateur est disponible (car il est, à la différence du supercalculateur, à la portée de n'importe quel scientifique) et standardisé. Il est en particulier primordial que les capacités graphiques de l'ordinateur jouent un rôle important dans

la pratique de la modélisation, parce que la lecture facilitée des résultats améliore la capacité à peaufiner le modèle instantanément. Un statut qui embrasse totalement la chimie computationnelle. Mais il est également favorisé par la diffusion des logiciels. Le logiciel de modélisation est l'artefact qui permet la diffusion de la nouvelle forme de modélisation.

Ainsi, ce logiciel est une marchandise à diffuser, ce qui renforce encore l'affirmation selon laquelle il agit comme une « technologie-recherche ». La marchandisation implique plusieurs caractéristiques énoncées par Shinn : les logiciels fonctionnent dans plusieurs disciplines, à travers le clivage public-privé, ils sont isolés des contextes de leur invention afin d'être largement distribués. Les logiciels, en tant que marchandises, sont des boîtes noires qui dépendent, en partie, de détails (paramètres ou routines) dont les mises en œuvre à l'intérieur du programme ne sont pas connues de l'utilisateur. La modélisation informatique est devenue elle-même une marchandise.

« Le mouvement des logiciels ainsi que l'adaptation locale [...] facilite l'échange d'informations scientifiques », affirment Lenhard et Johnsson. Leur description est généraliste, ils ne se concentrent pas sur un domaine scientifique précis. Pour d'autres domaines scientifiques, l'ère du *desktop modeling* s'est imposée plus tard, ou dans un autre contexte économique, académique et politique. Dans le cadre de la modélisation du climat, elle apparaît à une époque de plus grande ouverture aux standards ouverts. Dans le cadre de la bio-informatique, le *desktop modeling* a été la norme depuis le début de la discipline, plus récente. Pour ces deux domaines, les communautés scientifiques académiques ont épousé les pratiques caractéristiques de Kely dans la création de logiciels, souvent en harmonie avec des pratiques scientifiques inspirées du libre pour « fomentier des mouvements » comme l'*Open Science* ou l'*Open Data*. Le logiciel libre a représenté ainsi une norme suffisamment forte (au sens de Gambardella et Hall) pour que les logiciels de modélisation soient créés puis diffusés dans ce cadre. La chimie computationnelle est unique dans le sens où il s'agit d'un domaine scientifique apparu à l'époque des supercalculateurs, et d'une science financée par les investissements fédéraux (ou nationaux). Pour la chimie computationnelle, contrairement à la modélisation du climat, le *desktop modeling* est apparu à l'ère de la science vécue comme une entreprise, et a impliqué que les gens de la communauté ont dû s'adapter

dans un contexte violent et incertain, un contexte qui a généré de nombreuses tensions et de nombreuses stratégies différentes pour la distribution de logiciels. En ce sens, le passage au *desktop modeling* a été une « épreuve » pour les chimistes computationnels ¹.

7. La « générification » des logiciels

Cette vision de l'activité de modélisation a une conséquence : les logiques de diffusion impliquent que le logiciel de modélisation moléculaire soit conçu pour un marché, plutôt que pour un utilisateur. Cette idée est aussi une bonne définition de ce à quoi se réfère Shinn quand il décrit l'une des caractéristiques d'une « technologie-recherche » : elle est générique.

L'éternelle « crise du logiciel » décrite par Ensmenger, implique que les coûts de développement de logiciels sont de plus en plus élevés. Les entreprises qui commercialisent des logiciels scientifiques font facilement faillite (particulièrement quand l'industrie pharmaceutique est en crise, dans le cas de la chimie computationnelle) et celles qui survivent le font en élargissant leur champ d'application et leur base de clients, entre autres par fusion ou acquisition. Avec encore plus de clients à traiter, le coût de l'entretien et de la réactivité aux problèmes (comme la réalisation et l'actualisation d'un *fucking manual* par exemple) est encore plus exigeant. Neil Pollock et ses collaborateurs, en explorant des « biographies » de logiciels industriels, décrivent un processus qui convient parfaitement à cette situation : ils étudient la façon dont le logiciel évolue dans son processus d'acquisition de clients. Ils ne décrivent pas l'« appropriation » du logiciel par les différents utilisateurs locaux, mais plutôt la façon dont le logiciel est façonné de sorte qu'il puisse être utilisé aussi largement que possible ². Ils appellent ce processus une « générification » du logiciel, un terme qui renvoie à une des caractéristiques de la « technologie-recherche » employée par Terry Shinn mais aussi à la situation du logiciel

1. En référence à la « sociologie des épreuves ».

2. Neil POLLOCK, Robin WILLIAMS et Luciana D' ADDERIO, « Global Software and Its Provenance : Generification Work in the Production of Organizational Software Packages », dans : *Social Studies of Science* 37.2 (2007), p. 254–280.

de modélisation moléculaire dans un contexte de temps économiques difficiles.

La conception des logiciels « génériques » diffère radicalement des anciennes pratiques de développement de logiciels, particulièrement dans le domaine de la recherche scientifique. Dans ces dernières, les fournisseurs entretenaient traditionnellement des liens étroits avec les utilisateurs, selon l'idée qu'une meilleure connaissance des utilisateurs conduirait à une meilleure conception. À l'opposé, les fournisseurs de solutions « généralisées » gardent activement les utilisateurs à distance. Ils craignent que leur logiciel soit identifié et lié à des utilisations spécifiques, le rendant ainsi non commercialisable à grande échelle. Si l'idée est que le logiciel est conçu pour avoir l'étendue d'utilisation la plus large possible, le plus grand nombre d'utilisateurs possible, alors les fournisseurs évitent de traiter avec les utilisateurs individuels, pour minimiser leurs coûts de développements.

C'est exactement ce que les vendeurs de logiciels de modélisation moléculaire ont fait, en concevant des logiciels incluant le plus grand nombre de modèles possible, brouillant les frontières entre des modèles aussi différents que chimie quantique et mécanique moléculaire, dans une course à embrasser autant de clients que possible.

Le processus de « généralisation », selon Pollock et al., n'est pas simplement une tendance à devenir un logiciel polyvalent. Il consiste en une mécanique impliquant la communauté d'utilisateurs. Pollock et al. l'appellent « gestion de communautés » (*community management*) pour décrire le fait que l'évolution du logiciel a besoin de l'adhésion (*enrollment*) d'utilisateurs initiaux qui constitueront la base sur laquelle les développeurs de logiciels comptera, avec pour effet d'exclure le reste des utilisateurs (c'est-à-dire la majeure partie d'entre eux, plus tardifs, ou moins impliqués) du processus d'élaboration du logiciel.

L'idée est de constituer une communauté suffisante d'utilisateurs, puis de la façonner de sorte que ladite collectivité corresponde le plus possible aux capacités du logiciel. Dans un processus de façonnage mutuel (*mutual shaping*), cette « première couche » d'utilisateurs va essayer de façonner le logiciel à ses besoins particuliers. La gestion des communautés consiste donc à s'appuyer sur une petite fraction des utilisateurs pour écarter la majeure partie d'entre eux. Abandonnés par les développeurs,

le reste des utilisateurs posent de nombreuses questions sur les forums et les listes de diffusion (*mailing lists*) et les injonctions à lire le *fucking manual* qu'on y trouve traduisent les tensions générées dans la communauté par le sentiment d'abandon. Les utilisateurs de logiciels, mécontents de la façon dont ils sont traités par la maintenance officielle des vendeurs ou des développeurs se tournent massivement vers d'autres sources d'aide et de réconfort... avec plus ou moins de succès. Cette gestion des communautés est à mettre en regard de la quatrième pratique décrite par Kelty qui est la coordination de collaborations. Pour qu'un projet libre soit efficace, mettre en place des collaborations dans la communauté d'une façon ou d'une autre est important. On voit bien, par opposition, qu'au logiciel « générifié » correspond une stratégie sélective dans laquelle la gestion de la communauté consiste à écarter la majorité de celle-ci, et que cette stratégie est source de tensions, particulièrement si la communauté en question est une communauté scientifique, censée, selon ses normes mertonniennes communément admises, reposer sur l'universalité, voire le partage. Même s'il serait naïf d'invoquer l'éthique de Merton pour décrire la communauté scientifique (de même que Kelty préfère décrire des pratiques du libre plutôt que des normes), on ne peut que constater que la gestion de communauté du logiciel « générifié » implique des tensions dans la communauté scientifique, du à un choc avec les idées d'universalisme et de communalisme, a priori plus compatibles avec la quatrième pratique de coordination de collaborations.

Pollock décrit le logiciel générique comme un blob noir (*black blob*) au lieu d'une boîte noire (*black box*) : une entité amorphe, accessible seulement aux utilisateurs les plus proches et difficile à appréhender pour les utilisateurs les plus éloignés. L'ouverture (*openness*) des logiciels est ambiguë dans le cadre de cette « générification » : dans un premier temps, ouvrir le code source est une stratégie utile pour enrôler des utilisateurs dans une communauté. Puis, quand le logiciel se développe, l'ouverture devient un inconvénient pour le processus, précisément parce qu'il est considéré comme contre-productif que les utilisateurs soient trop au fait de la machinerie du logiciel. Cette idée est encore à mettre en regard de la première pratique du libre de Kelty : partager le code source. Il est bien sûr évident que le partage du code source est l'essence même du libre, mais il est encore intéressant de noter que dans notre cas, la question est

un problème et encore une fois une source de tension. Pour un logiciel scientifique commercial, la tension existe aussi entre la tendance à protéger son code de la concurrence (pour éviter le *reverse engineering*) et le souhait de la communauté d'accéder à l'intérieur de la boîte noire au nom de la reproductibilité scientifique.

À cet égard, les solutions imaginées par les chimistes computationnels sont multiples, mais l'un d'entre eux, le best-seller en chimie quantique, a développé une idée unique : rendre le code source disponible, mais pas ouvert : une initiative rare d'un logiciel sous licence propriétaire avec code source partagé. Cette approche surprenante a été la source des plus hautes tensions dans la communauté d'autant plus que la licence du logiciel interdit expressément aux utilisateurs de comparer (*benchmark*) et de rendre publiques les performances du logiciel.

La saga du logiciel en question, de ses licences, de ses procès, depuis les années quatre-vingt jusqu'à aujourd'hui est édifiante. Elle résume les tensions et les controverses à l'intérieur de la communauté entre développeurs et utilisateurs. D'autres logiciels ont imaginé de nombreuses autres stratégies, de nombreux autres *business models*. De licences annuelles renouvelables en produits d'appel gratuits mais propriétaires, code source ouvert ou code source fermé, publié ou non, aucune solution ne s'est vraiment imposée comme une norme au sein de la communauté scientifique, révélant la diversité des pratiques et la difficulté à les rendre compatibles entre elles. La troisième pratique décrite par Kelty, l'écriture de licences, prend ici tout son sens. Qu'on soit dans le cadre du libre, ou en train de développer un logiciel « générifié », c'est par l'écriture de la licence que les stratégies se développent (et s'analysent), et la multitude de choix différents dans le domaine de la chimie computationnelle illustre bien les incertitudes stratégiques dans le domaine de création de logiciels en chimie computationnelle, à l'ère du *desktop modeling*.

8. What FM ? Tensions dans la communauté

Ce qui s'est passé avec le logiciel de modélisation moléculaire au cours des années quatre-vingt et quatre-vingt-dix est unique. Le virage du *desktop modelling* en tant que rupture dans l'activité scientifique a été une

source de tensions chez les chimistes computationnels qui ont été les premiers à recevoir de plein fouet ce tournant. Ce qui est arrivé est caractéristique d'un contexte politique, économique, technologique et scientifique mais aussi une aventure de création de logiciels unique. La communauté (dans sa diversité et ses tensions) a façonné le logiciel, et en retour celui-ci a transformé l'activité scientifique, à l'image de la communauté, dans un contexte de promesse économique puis de crise et en l'absence de normes : les voies explorées par les développeurs reflètent ces tensions.

Les nombreuses solutions (parfois uniques) trouvées par les développeurs de logiciels de calcul de chimie pour survivre dans le monde du *desktop modelling* ont été conceptualisées grâce à la notion de « généralisation », amenés d'une part par Shinn (et Lenhard) et de l'autre par Pollock.

Les pratiques caractéristiques des communautés du libre de Kely ont été comparées aux choix effectués dans la chimie computationnelle. Il est frappant de constater comme les cinq pratiques nécessaires à forger une communauté sont autant de tensions dans le domaine de la chimie computationnelle... une communauté loin du libre.

L'ouverture du code source, tout d'abord, est loin d'être naturelle. Elle est loin de faire l'unanimité dans le monde académique des années quatre-vingt, d'une part aux yeux des administrations universitaires pour lesquelles le « transfert de technologie » est plus important que des valeurs abstraites comme l'universalité de la science, d'autre part aux yeux des développeurs de logiciels commerciaux qui ressentent une faible reconnaissance de la part de la communauté scientifique.

L'idée d'un système ouvert, à cet égard, pour les développeurs de logiciel, est donc vue comme un avantage éventuel au début du développement d'un projet (pour enrôler une communauté d'utilisateurs rapprochés) mais devient vite un défaut quand, au contraire, il faut gérer une communauté d'utilisateurs grandissante, multiple et diversifiée, dans le cadre de la « généralisation » du logiciel, logiciel vu comme une marchandise.

La communauté (d'utilisateurs, de développeurs) n'est donc pas forcément un endroit où l'on collabore. Le processus de *community management* des logiciels en voie de « généralisation » consiste plutôt à diviser les utilisateurs, afin que le logiciel de plus en plus polyvalent soit utilisé par le

plus grand nombre plutôt qu'approprié de manières diverses par diverses communautés d'utilisateurs.

Il n'y a donc pas non plus de mouvement, ni d'idée globale et partagée de ce que doit ou devrait être la modélisation moléculaire ou la chimie computationnelle. Une absence de normes, ou d'imaginaires communs, couplée à la demande du marché au moment où cette science a pris son essor ainsi qu'aux perspectives de profits à cette époque, représente les conditions typiques pour empêcher le succès du logiciel libre dans un milieu académique.

Références

- ENSMENGER, Nathan L., *The Computer Boys Take Over : Computers, Programmers, and the Politics of Technical Expertise*, Cambridge : MIT Press, 2010.
- FLICHY, Patrice, « Internet ou la communauté scientifique idéale », dans : *Réseaux* 17.97 (1999), p. 77–120.
- FRANCOEUR, Eric et Jérôme SEGAL, « From model kits to interactive computer graphics », dans : *Models : The Third Dimension of Science*, sous la dir. de Soraya de CHADAREVIAN et Nick HOPWOOD, Palo Alto, 2004.
- GAMBARDELLA, Alfonso et Bronwyn H. HALL, « Proprietary versus public domain licensing of software and research products », dans : *Research Policy* 35.6 (2006), p. 875–892.
- JOERGES, Bernward et Terry SHINN, *Instrumentation Between Science, State and Industry*, Boston : Kluwer Academic Publishers, 2001.
- JOHNSON, Ann et Johannes LENHARD, « Toward a new culture of prediction : Computational modeling in the era of desktop computing », dans : *Science Transformed ? Debating Claims of an Epochal Break*, sous la dir. d'Alfred NORDMANN, Hans RADDER et Gregor SCHIEMANN, Pittsburgh : University of Pittsburgh Press, 2011, p. 189–200.
- KELTY, Christopher, *Two bits : the cultural significance of free software*, Durham : Duke University Press, 2008.
- MOWERY, D. C. et al., « The growth of patenting and licensing by US universities : an assessment of the effects of the Bayh-Dole act of 1980 », dans : *Research policy* 30.1 (2001), p. 99–119.

POLLOCK, Neil, Robin WILLIAMS et Luciana D'ADDERIO, « Global Software and Its Provenance : Generification Work in the Production of Organizational Software Packages », dans : *Social Studies of Science* 37.2 (2007), p. 254–280.

RAYMOND, Eric S., *The Cathedral and the Bazaar*, Sebastopol, CA : O'Reilly, 2001, URL : <http://www.catb.org/esr/writings/homesteading/>.

VI

Témoignages : Libre
et institutions

Un point d'histoire de la stratégie d'Inria sur *l'open source*

Gérard GIRAUDON

Comme Monsieur Jourdain qui faisait de la prose sans le savoir, l'IRIA puis Inria, diffusait du logiciel en code ouvert depuis la fin des années 1970. Cela a bien sûr été facilité par l'arrivée de l'application FTP sur l'Internet à Inria au milieu des années 1980 et largement amplifié grâce à la première connexion directe TCP/IP entre la France et les États-Unis, réalisée en juillet 1988 entre le centre Inria à Sophia Antipolis et l'université du Wisconsin à Madison. En fait, pour être tout à fait exact, il n'y avait pas de position bien affirmée au niveau de la direction de l'institut et cette diffusion était globalement laissée à la décision des chercheurs des équipes qui l'utilisaient comme de nombreux collègues à travers le monde pour échanger avec leur réseau de pairs, à l'instar de ce que l'on fait avec des publications¹. Cette diffusion concernait le logiciel vu comme un objet de connaissance et de recherche permettant de faire avancer la science que l'on appelle aujourd'hui la Science du Numérique. La stratégie de diffusion du logiciel en code ouvert vu comme un objet de transfert technologique restait à construire².

1. On peut citer par exemple l'initiative de Pierre Weiss au début des années 1990 de l'édition d'un CD-Rom des logiciels libres de l'INRIA Rocquencourt.

2. Il faut insister ici sur la caractéristique unique du logiciel dans l'histoire des sciences et techniques : tout à la fois objet de connaissance transmissible par l'écrit et objet techno-

En 1994, sous l'impulsion de son PDG de l' poque, Alain Bensoussan, et de son Directeur scientifique, Gilles Kahn, Inria  labore son premier plan strat gique. On peut y lire quatre principes fondateurs ¹ :

1. La n cessit  de l'excellence scientifique au meilleur niveau mondial coupl e   une vision affirm e sur le transfert technologique et l'innovation guid e par une forte volont  d'impact.
2. Le constat est que si la domination scientifique passe naturellement par de grands m dias de communications scientifiques (conf rences, soci t  savantes), elle passe aussi en informatique par une strat gie « *astucieuse* de mise dans le domaine public de logiciels ou de technologies qui s'imposent alors comme des standards de fait. En fait la r volution des syst mes ouverts qui a tant boulevers  l'industrie informatique [...] trouve son origine en grande partie dans cette cr ation de standards de fait par la communaut  scientifique am ricaine. »
3. « La num risation et le r le du logiciel deviennent pr dominants [...] l'informatique appara t comme un  l ment cl  de la comp titivit  de tous les secteurs  conomiques, et m me de l' volution de la soci t  [...] La r ponse passe par la mise en place de syst mes complexes bien organis s, faciles   utiliser int grant de nombreuses couches ou briques de bases. »
4. L'importance du transfert par la cr ation d'entreprises innovantes illustr e par la performance de l'industrie du logiciel am ricain gr ce au mod le start-up.

Ces principes allaient structurer l'action future de l'institut jusqu'  nos jours.

C'est ainsi que, d s 1995, l'institut co-fondait avec le MIT la branche europ enne du W3C,  tant convaincu de la r volution que le Web pouvait apporter si la pr servation de son interop rabilit  par des standards ouverts  tait assur e.

Concernant l'*open source*, l' laboration d'un plan d'action a  t  plus lente.   la vision premi re de la diffusion de logiciel *open source* comme

logique une fois compil  sur un ordinateur et dont le passage de l'un   l'autre est quasiment   co t nul.

1. <http://www.inria.fr/institut/strategie/plan-strategie>. PS 1994-1998, pp. 3-4-5

diffusion de connaissance, restait posé le questionnement sur la manière dont la diffusion de logiciel libre et ouvert permettait de créer un effet de levier et un véhicule pour le transfert technologique et la création de valeur. Le premier document qui traite de l'importance de la diffusion de logiciel *open source*¹ a été diffusé au comité de direction de l'institut en avril 1995. Il décrit les avantages de ce mode de diffusion au regard des quatre principes édictés ci-dessus et illustre le propos avec quelques exemples (projet GNU de la Free Software Fondation, bibliothèque de traitement d'image Khoros de l'université de New Mexico et son impact notamment pour la formation, le logiciel Ptolemy de l'université de Californie à Berkeley et son impact vers les partenariats avec des grands groupes industriels). La démonstration mettait en valeur le cercle vertueux² qui pouvait en découler entre l'impact sur la notoriété scientifique et technologique, la création de la référence et la création de la demande issue du marché, l'impact sur la collaboration entre public et privé par le rôle de facilitateur du logiciel *open source*. Ainsi le texte proposait de déployer une stratégie d'action selon la nature du logiciel (langage, environnement, bibliothèque, générique ou spécifique), sa cible (enseignement, recherche, entreprises), sa qualité et sa facilité d'usage, etc. Le document concluait sur la nécessité d'une politique affirmée, adossée à l'élaboration de critères explicites pour faire des choix (*open source* ou pas) et le besoin d'une meilleure mise en œuvre (centralisation de l'information, choix d'une licence, identification des auteurs, etc.). Ce texte a permis de lancer le débat stratégique au sein de la direction mais sans qu'une politique affirmée soit arbitrée.

L'année 1996 est l'année du changement de PDG (arrivée de Bernard Larrouturou), la création d'une fonction de Délégué Général au Transfert et la création de la Direction du Développement et des Relations Industrielles. La priorité était de mettre en œuvre la stratégie autour de l'aide à la création d'entreprises innovantes qui était élaborée depuis 1994 sous l'impulsion de Jean-François Abramatic et de Laurent Kott. Cela a abouti à la création d'INRIA-Transfert et des fonds d'amorçage portés par ISourceGestion à partir de 1998.

1. Le titre exact était : « Réflexion autour des FTP anonymes pour la diffusion de logiciel » document daté du 3 avril 1995. G. GRAUDON (archives de l'auteur).

2. Il est fait référence à la fertilisation croisée de la trilogie Enseignement-Recherche-Industrie chère à Pierre Laffitte, fondateur de la technopole Sophia Antipolis.

Concernant l'*open source*, l'ann e 1998 contient  galement une date importante. En effet, le 20 mars 1998,   l'occasion de la f te de l'Internet, Inria organise   la Mutualit    Paris une conf rence-d bat sur le th me « Logiciels libres : impact  conomique », avec des discours d'ouverture de Bruno Oudet et de Gilles Kahn. On notera pour l'histoire l'intervention de M. Chatelain pour Netscape et de M. Young pour une jeune entreprise encore inconnue du large public, Red Hat. C'est ce jour-l , dans ces locaux, que l'AFUL a pris naissance, association   laquelle participa un Directeur de Recherche Inria bien connu de la communaut  du logiciel libre en France, Bernard Lang. 1998 est  galement l'ann e de la diffusion au Comit  de direction Inria d'une deuxi me version du texte de 1995. Si l' laboration d'une strat gie claire sur l'*open source* tardait   sortir depuis lors, c'est que la question, objet du colloque mentionn  ci-dessus, restait totalement ouverte. Si des mod les  conomiques commen aient   s' laborer, aucun aux yeux de l'institut ne permettait d'assurer une p rennit  des entreprises et des emplois en France, voire en Europe, compte tenu du contexte industriel, de la difficult    monter une industrie du logiciel en France, cons quence du peu d'app tence des grandes entreprises ou des pouvoirs publics fran ais pour l'innovation venue de France ¹.

Il  tait clair pour Inria que la cl   tait dans l' laboration d'un  cosyst me «  veill  », dont les composants  taient identifi s :

- communaut  de chercheurs et communaut  de d veloppeurs ;
- entreprises de l'offre technologique (briques, composants) ;
- entreprises de services aptes   customiser une solution client et   la maintenir et
- surtout les clients finaux et donneurs d'ordres aptes   exploiter les solutions (entreprises, collectivit s, directions de l' tat).

Le verrou, nous semblait-il,  tait la cr ation de la confiance ; la confiance de chacun des acteurs de l' cosyst me dans l' cosyst me lui-m me, au-del  de la confiance dans les autres acteurs. Nous allons voir ci-apr s comment Inria a contribu    lever les obstacles mais dans tous les cas, nous consid rions que la cr ation de la confiance se construisait

1. Inria avait acquis de l'exp rience   travers ses *spin-off* qui se pr sentaient comme  diteurs Logiciel, et notamment ses filiales de l' poque comme Simulog, Ilog, Connexit  etc.

par la prise de conscience de la part de chacun des acteurs que sa contribution à l'écosystème lui-même était un investissement rentable pour lui mais aussi pour l'écosystème ; ce que nous illustrons ici par l'adjectif « éveillé ». Cet adjectif signifie que chaque composante de l'écosystème est consciente qu'elle reçoit de l'écosystème mais qu'elle doit également y contribuer.

Inria était convaincu que les écosystèmes *open source* pérennes étaient un formidable changement de paradigme¹, mais ils ne pouvaient pas être une table ouverte où certains ne font que s'asseoir pour s'y restaurer. Enfin, Inria était convaincu que le succès en matière de transfert et d'emploi irait aux acteurs économiques les plus rapides, ceux aptes à exploiter les nouvelles opportunités et aptes à prendre des risques.

Début 2000 s'engage une nouvelle réflexion pour faire avancer la stratégie de l'institut sur le développement et le transfert. Cette réflexion aboutit en 2001 à un texte fondateur présenté au Conseil d'administration d'Inria sur le transfert et les axes de mise en œuvre et l'organisation du développement technologique à l'institut à travers des services explicites. La diffusion de logiciels *open source* est mentionnée comme un axe important et notamment à travers la création de consortiums. Cela se traduira peu après par la création du Consortium ObjectWeb², la création du Consortium Scilab, la création du Consortium Caml ou encore la forte participation à des initiatives comme IHE (Integrating the Healthcare Enterprise).

En parallèle, mais en retard par rapport à leurs homologues américaines, les entreprises françaises et européennes à logiciel prépondérant s'intéressent de plus en plus à l'*open source*, à son développement et aux écosystèmes qu'il permet de créer. C'est ainsi qu'en novembre 2001, est diffusé un rapport RNTL sur le logiciel libre³ et sept ans plus tard le Cigref publie un rapport de synthèse sur l'*open source*⁴.

1. Voir Gérard GIRAUDON, « Le logiciel libre et ouvert : révolution ou évolution ? » dans *Interstices*, 2006. URL : http://interstices.info/jcms/c_14658/le-logiciel-libre-et-ouvert-revolution-ou-evolution.

2. Objectweb a donné naissance en 2006 à OW2.

3. Le GT du RNTL qui a écrit ce rapport était composé de : Alain Conchon (co-animateur), Jean-Michel Dalle, Gérard Giraudon (co-animateur), Bernard Rougeot, Jean-Paul Smets, Jean-Michel Tanneau (rapport dans les archives de l'auteur).

4. Voir la synthèse téléchargeable sur la page <http://www.cigref.fr/lopen-source-en-entreprise>.

Comment en est-on arriv   l   ?

Comme indiqu   pr  c  demment Inria   tait depuis 1995 membre fondateur du W3C, consortium international qui   mettait avec le succ  s que l'on conna  t des recommandations sur les sp  cifications d'  volution du Web pour en assurer l'interop  rabilit   et l'universalit  . S'appuyant sur sa propre exp  rience Inria a alors consid  r   que la bonne strat  gie   tait de coupler standard et *open source*. De plus, Inria   tait convaincu que dans certains domaines o   l'interop  rabilit     tait une condition premi  re du d  veloppement et une « commodit   » pour chacun, le logiciel libre   tait strat  gique car le logiciel devenait lui-m  me le standard ¹. La vision Inria   tait que les utilisateurs finaux devaient   tre au c  ur de ces consortiums car, outre leur capacit      exprimer suffisamment en amont leurs besoins, ces utilisateurs finaux devaient jouer le r  le cl   du « marketing-commercial ». En effet, il   tait clair que le mod  le   conomique ² d'un   cosyst  me devait   tre viable s'il voulait durer. Un des int  r  ts financiers du mod  le *open source* pour un utilisateur final est qu'il ne « paye » que pour la partie terminale du produit, celle qui correspond    son besoin personnel ;    ce titre le mod  le financier est extraordinaire puisque l'on peut faire du sur-mesure au prix du pr  t-  -porter. Mais si on trouve dans un   cosyst  me *open source* les composantes pour mettre en   uvre la recherche, le d  veloppement, la documentation, la sp  cialisation, on ne trouve personne pour assurer le marketing et le commercial de l'  cosyst  me lui-m  me. Ainsi, le r  le cl   des utilisateurs finaux   tait celui-l  , c'  st-  -dire celui de faire savoir leur satisfaction (ou pas) en enclenchant ce que l'on nomme maintenant le marketing viral. Or, force a   t   de constater que ce n'  st pas vraiment ce qui s'  st pass  , du moins en France. En effet, un certain nombre d'obstacles se dressaient sur le chemin, culturel en premier bien s  r, la notion de symbiose n'  tant pas toujours une valeur des utilisateurs finaux, mais aussi de confort (l'acte d'achat ayant valeur de solde de tout compte voire un g  ge de minimisation des risques vis-  -vis de sa hi  rarchie).

1. En effet, le logiciel est    la fois la sp  cification et son impl  mentation et parce qu'il est ouvert, chacun peut le lire comme une sp  cification et s'interfacer dans la mise en   uvre.

2. Par   conomie, il faut entendre ici l'acception g  n  rale du mot, c'  st-  -dire l'ensemble des conditions   conomiques et pas seulement celles qui se traduisent par un flux financier. On peut dans ce registre citer les travaux issus de financement RNTL de J. M. DALLE et N. JULLIEN publi  s en 2003, « Libre Software : Turning fads into institutions ? », dans *Research Policy*, 32/1, pp. 1-11. URL : <http://www.marsouin.org/spip.php?article79>.

Parmi ces obstacles, il y en avait un qui nous semblait pouvoir être levé. Il s'agissait du risque juridique. En effet, même si au sein de nombreuses entreprises, dont certains grands groupes, l'utilisation de l'*open source* était présente¹ et même si on trouvait des fervents partisans de la notion d'écosystème symbiotique, la plupart des grandes décisions passaient par un avis sur le risque juridique. Or, au début des années 2000, l'analyse d'Inria était que, d'un point de vue strictement juridique, très peu de licences étaient un vrai contrat entre les parties (aucune mention de droit de référence en cas de litige et celles qui en mentionnaient étaient la plupart nord-américaines). Ce constat était un obstacle d'autant plus important que le logiciel prenait, comme on l'avait prévu au début des années 1990, un rôle clé au sein de l'entreprise voire celui de vrai différenciateur vis-à-vis des concurrents ; le rôle de la propriété intellectuelle concernant les logiciels devenant de plus en plus important. Fort de ce constat et parce que nous étions aussi demandeurs de sécurité juridique pour les logiciels diffusés par nos établissements, un petit groupe composé de scientifiques d'Inria et de juristes du CEA et du CNRS se sont lancés en septembre 2003 dans la rédaction de la famille des licences CeCILL². Ce petit attelage avait pour but de faire converger relativement rapidement sur trois licences en adaptant au droit européen, incarné via ses directives dans le droit français, l'esprit des trois licences les plus utilisées pour la diffusion *open source* : la GNU GPL, la GNU LGPL et la BSD³. Nous en avons également profité pour prendre en compte les dernières tendances de l'informatique. C'est ainsi que la CeCILL fait référence à des brevets possibles, la CeCILL C essaye de clarifier la césure de l'hérité de la licence notamment pour les bibliothèques et la CeCILL B instaure un devoir de citation. Ce dernier point nous semblait concilier l'intérêt gagnant-gagnant entre un utilisateur qui peut récupérer un logiciel diffusé sous cette licence dans tout ou partie de ses propres produits et l'intérêt de l'auteur de pouvoir être cité à l'instar d'une citation d'une publication⁴.

1. Linux ou les serveurs Apache en sont les archétypes.

2. Pour Cea, Cnrs, Inria Logiciel Libre : <http://www.cecill.info>.

3. Un quatrième type de licence avait été prévu initialement (il s'agissait de la CeCILL D pour Data) mais n'a finalement pas vu le jour par manque de temps avant que le groupe ne se disperse ; depuis est sortie la licence de droit français données ouvertes : <http://www.data.gouv.fr/Licence-Ouverte-Open-Licence>.

4. L'idée d'un template XML a même été soumise par un industriel qui nous servait de bêta-testeur.

Cela devait permettre   des chercheurs de pouvoir montrer,   la mani re des indices bibliom triques de leurs publications, l'impact de leur logiciel et ainsi conforter leur dossier de carri re. Cela devait r soudre en partie la tension  voqu e au d but de l'article entre notori t  (logiciel objet de connaissance) et cr ation de valeur (logiciel objet de transfert) y compris au sein d'un logiciel « ferm  » si c' tait la meilleure voie pour innover et cr er des emplois.

Sur cette lanc e, en 2009, Inria a  tabli un livre blanc intitul  *Strat gie d'Inria sur le logiciel libre*. Par les positions affirm es dans ce texte, l'institut r affirmait son soutien   la dynamique du logiciel libre, jug  indispensable au d veloppement de la soci t  num rique de la connaissance. Cette prise de position met l'accent sur l'utilisation du logiciel libre comme un outil de transfert. Sont pr cis es les conditions dans lesquelles le recours au logiciel libre est   m me de maximiser l'impact des logiciels d velopp s   l'institut, au-del  du cercle de la recherche publique fran aise et europ enne, en inventant un nouvel espace de transfert.

Si naturellement le logiciel libre est un objet de diffusion, son enseignement, sa conception (nouvelle approche de la programmation, d veloppement communautaire) et sa distribution (release, etc.) posent de nouvelles probl matiques. Sur l'initiative de l'institut et sous la houlette de Roberto di Cosmo, Inria et les universit s Pierre et Marie Curie et Paris-Diderot ont fond  en 2010 l'IRILL (Initiative pour la Recherche et l'Innovation sur le Logiciel Libre). En effet, objet aux dimensions multiples, le logiciel libre n cessite de croiser des approches et des comp tences diverses et le Centre IRILL permet de r unir des acteurs de la recherche, de la formation, du transfert et de l'innovation, dans la dur e et dans un lieu identifi    Paris.

Ensuite, Inria a pris l'initiative de promouvoir le logiciel libre au travers de manifestations comme fOSSa (Free Open Source Software Academia) et plus r cemment le concours « Boost your Code » qui offre   un tout jeune dipl m  une ann e de salaire pour d velopper son projet personnel *open source*.

Enfin, et dans la ligne directe du logiciel *open source*, une nouvelle dimension se construit autour des probl matiques de l'*open data* qui est riche de promesse et o  l'interop rabilit  aura  galement un r le majeur.

Après bientôt deux décennies de réflexion et d'action, et la transformation du monde par la révolution de l'informatique et des sciences du numérique, l'histoire du logiciel libre et ouvert – et maintenant des données ouvertes – ne s'arrête pas car nous sommes toujours convaincus qu'au-delà des clivages idéologiques, la dynamique du logiciel libre et ouvert facilite la création d'un dialogue permanent entre recherche, industrie et société.

Remerciements

L'auteur tient à remercier P. Moreau, S. Ribas, S. Ubeda et C. Zaetta pour leur conseil et leur relecture.

La place du logiciel libre au CNRS

Témoignage d'un ingénieur en informatique au CNRS depuis 1984

Jean-Luc ARCHIMBAUD

Cet article, rédigé en août 2012, apporte un témoignage sur presque 30 ans de vie professionnelle comme informaticien au Centre National de la Recherche Scientifique (CNRS). Son but est d'aider à comprendre pourquoi aujourd'hui les logiciels libres sont présents ou non dans les laboratoires de recherche et comment ils le sont.

Ingénieur en informatique diplômé de l'INSA de Lyon, j'ai d'abord travaillé comme ingénieur développeur pour la société BULL pendant trois ans. Je suis ensuite entré au CNRS en 1984, comme ingénieur système et réseau, et j'y suis encore avec un profil actuel de responsable de projets. Le CNRS, organisme de recherche public, regroupe plus de 1000 laboratoires, dans toutes les disciplines scientifiques. Ses spécificités en font un lieu où l'on peut avoir une vision sur l'ensemble des pratiques de recherche, dans toutes les sciences.

J'ai travaillé six années durant dans un centre de calcul pour les universités et les laboratoires, le Centre Inter-universitaire de Calcul de Grenoble (CICG), sur le campus de Grenoble, ce qui m'a permis d'avoir un contact direct de service avec des chercheurs et une vision universitaire de la recherche. Puis de 1990 à 2010, j'ai travaillé dans une unité nationale,

l'Unité Réseaux du CNRS (UREC)¹, chargée en particulier de la mise en place des réseaux et de leurs applications dans l'ensemble des laboratoires du CNRS. Cette mission m'a permis de visiter (et de conseiller dans mes domaines de compétences) de très nombreux campus et laboratoires, dans tous les domaines de recherche.

Mes activités et compétences techniques ont évolué dans le temps : développement de logiciel et connectivité des premiers PC, infrastructures de réseaux informatiques (réseaux locaux, de campus, nationaux), technologies IP, applications internet, sécurité informatique, grilles de calcul, référencement de logiciels libres (plate-forme PLUME²), qualité, portails thématiques. J'ai monté et piloté de nombreux projets ou groupes de travail. Je ne suis pas un développeur logiciel, mais j'ai fait (ou coordonné) de nombreux tests, référencements, recommandations sur des logiciels, et ce, pour l'ensemble des laboratoires.

Durant ces années professionnelles, j'ai assisté, parfois participé, à l'évolution des pratiques informatiques dans les laboratoires de recherche. Je citerai : l'arrivée des ordinateurs personnels, la mise en place des réseaux (locaux, de campus, nationaux), le déploiement des applications internet, la problématique de la sécurité, la généralisation du calcul et de la modélisation dans la recherche... Outre ces aspects techniques, les méthodes de partage des connaissances, la création et la vie de nombreux groupes d'échanges techniques avec les outils de travail collaboratif ont constitué un autre volet important de mes activités. Cette évolution de l'outil informatique et des pratiques dans les laboratoires s'est faite avec l'utilisation, parfois le développement, de logiciels libres ou non.

Je viens apporter un témoignage sur certains épisodes auxquels j'ai participé dans ma vie professionnelle en lien (parfois uniquement de ressemblance) avec le logiciel libre, à la fois dans son objet (source diffusée...) mais aussi dans ses pratiques (communauté vivante...). J'essaierai de prendre du recul pour en tirer une leçon, une vision plus globale. Je me limiterai à ce que j'ai vécu, côtoyé de très près. De nombreux faits beaucoup plus importants seront donc occultés. C'est ainsi une vision et

1. Voir Jean-Luc ARCHIMBAUD, « UREC 1990-2010, acteur majeur dans la mise en place des réseaux informatiques au CNRS », URL : <http://hal.archives-ouvertes.fr/docs/00/64/39/33/PDF/UREC.pdf>.

2. <https://www.projet-plume.org>.

une analyse tout à fait personnelles et donc incomplètes et partiales. J'en avertis le lecteur. Il est évident que cela n'engage pas le CNRS.

Ce témoignage est aussi un témoignage du terrain, donc avec un vocabulaire simplifié, celui qu'il faut adopter avec les décideurs pour ne pas les noyer dans les subtilités du monde du libre. Je n'emploierai ainsi que les termes « logiciel libre » ou « le libre » sans indiquer la définition précise.

Quand on lit un témoignage, il est important de savoir qui parle. Chacun possède sa vision en fonction de son vécu et de sa place dans l'histoire. C'est ce qui m'a conduit à décrire mon parcours professionnel en préambule, pour que vous sachiez à qui vous avez affaire.

Néanmoins, mon objectif n'est pas de « raconter mon histoire », ce qui a peu d'intérêt, mais de rapporter certains faits de ces trente dernières années qui peuvent aider à comprendre la place actuelle du logiciel libre dans les laboratoires de recherche. Comme vous le verrez dans certains paragraphes, il est difficile pour un ingénieur comme moi, personnel technique, d'influer sur la stratégie informatique, logiciel libre ou pas, d'une grande maison comme le CNRS. D'abord parce qu'il n'y a pas de stratégie ou alors plusieurs, une par direction, donc pas de lieu où argumenter. Néanmoins, un moyen d'influer est, d'un côté, de montrer que des logiciels libres de qualité existent et qu'ils sont largement utilisés, et d'un autre côté, de montrer que le modèle de développement et de diffusion du libre est performant et peut être un modèle appliqué à l'organisme. Ceci sans dénigrer systématiquement les logiciels commerciaux.

Cet article est une succession de petites histoires dans l'ordre chronologique et se termine par un essai de conclusion personnelle sur la position du libre dans la recherche.

Quelques faits passés

Travailler avec un système d'exploitation dont les sources sont disponibles et modifiables (1984-1986)

À mon arrivée au Centre Inter-universitaire de Calcul de Grenoble (CICG) en 1984, ce dernier était équipé d'une grosse machine fonctionnant avec le système d'exploitation Multics. Sur cet équipement, unique

outil informatique des laboratoires du campus, entre 50 et 100 chercheurs et enseignants étaient connectés en simultané, avec des terminaux non intelligents (le PC n'existait pas). Multics venait du MIT et de nombreux concepteurs de Multics ont travaillé ensuite à la création d'Unix, version plus simple de Multics. Ce système était écrit entièrement en langage PL1 (langage structuré très lisible) et nous disposions de toutes les sources que nous pouvions modifier et ré-utiliser à des fins de développement. Ceci était naturel, tous les systèmes informatiques utilisés dans la recherche étaient ainsi (le logiciel était alors quasiment gratuit).

Lorsqu'il y avait un problème délicat, un ingénieur système allait lire le code source pour en comprendre la raison. On pouvait ainsi étudier « comment ça marche ». Par cette pratique, les connaissances techniques de chacun en étaient augmentées. De plus, adapter le système à une spécificité française (le minitel par exemple) ou à un périphérique non prévu, n'était pas bloquant : on modifiait certains modules du système. Cet environnement était très formateur, il permettait de progresser techniquement et de développer du code rapidement.

Quand on a travaillé ainsi, on comprend tout à fait ce qui a conduit Richard Stallman à se lancer dans la définition du logiciel libre puis dans sa croisade pour son utilisation. Comme il est raconté dans sa biographie¹, chercheur en informatique à cette époque, il avait accès aux sources. Or, brusquement, il s'est vu interdire cet accès. Pour tous les chercheurs en informatique et plus globalement les ingénieurs en informatique, ce manque de visibilité constitua un frein très important dans le travail, une régression.

Participer à un groupe d'échanges techniques (1984-1986)

Multics était très peu répandu, une dizaine d'exemplaires en France, principalement pour la recherche publique et un peu privée. Le support technique du constructeur était ainsi *a minima*. Il fallait donc « se débrouiller seul ». Pas tout à fait, car les sites français qui possédaient un système Mutics s'étaient organisés pour créer des groupes d'échanges techniques. Ils se réunissaient régulièrement pour exposer les problèmes

1. Sam WILLIAMS, Richard STALLMAN et Christophe MASUTTI, *Richard Stallman et la révolution du logiciel libre. Une biographie autorisée*, Paris : Eyrolles, 2010.

de certains, échanger des pratiques et, souvent, des petits développements logiciel. Même sans Internet et sans les outils de communication et de travail collaboratif actuels, les échanges étaient très fructueux pour avancer ensemble, s'entraider. Il me semble que les rencontres toujours en présentiel, où l'on se côtoyait et dînait ensemble, permettaient de vraiment se connaître, ce qui évitait les échanges par courriel parfois durs dans des listes électroniques, qui peuvent souvent saboter l'ambiance d'un groupe de travail. Personnellement, c'était toujours avec plaisir que je participais à ces rencontres, même si les déplacements en train étaient longs.

Ces pratiques de groupes d'échanges n'étaient pas spécifiques à Multics, les autres systèmes ou constructeurs comme DEC, de mémoire, avaient leur équivalent. Ainsi, les échanges et le partage techniques, pratiques à la base du développement du logiciel libre et facilitées par l'utilisation de ces logiciels, ont existé dans la recherche dans le domaine informatique (ainsi que dans d'autres domaines), avant l'invention du Libre.

Distribuer gratuitement un logiciel sans licence, ni Internet (1986)

Multics avait un outil de messagerie local, identique au *mail* d'Unix d'ailleurs. Or, dans le monde de la recherche, beaucoup de sites étaient équipés de matériel IBM. IBM avait financé la mise en place d'un réseau académique BITNET aux États-Unis, basé sur des protocoles de communication propriétaires, et son équivalent européen, EARN, supporté aussi par le ministère de la Recherche. Il rendait de grands services en particulier pour la messagerie internationale entre les chercheurs. Les sites Multics en étaient exclus. C'est pourquoi je m'étais lancé dans le développement d'une passerelle entre la messagerie locale Multics et EARN. Grâce à l'accès aux sources en PL1, le travail avait été assez rapide et le résultat opérationnel.

Les autres sites Multics m'avaient demandé le code. Pour le diffuser, nous avons réalisé des copies du logiciel sur des bandes magnétiques et j'avais organisé une journée de présentation à Grenoble avec distribution de ces supports. Ainsi, j'avais fait une distribution d'un logiciel gratuitement sans licence et sans Internet. Certains sites avaient ensuite découvert des bugs et m'avaient proposé des corrections... Ceci pour montrer que l'objet *logiciel libre* existait avant la popularisation du terme, ainsi que

les pratiques de diffusion et d'amélioration des logiciels libres dans les laboratoires de recherche. Malheureusement, peu de temps après, l'arrêt du développement de Multics était annoncé, ce qui a limité la diffusion de mon développement à la France. Autrement, j'aurais peut-être fait de beaux voyages...

La simplicité et l'environnement international, deux critères importants dans les choix de solutions techniques (1990-1991)

Arrivé à l'Unité REseaux du CNRS, en 1990 j'ai participé activement à la définition puis à la mise en place du réseau informatique national de la recherche et de l'enseignement supérieur RENATER¹, piloté par le ministère de la Recherche et de l'enseignement supérieur. Au départ de la réflexion s'est posée la question du choix des protocoles de communication qui devaient circuler sur ce réseau. À l'époque, il y avait entre les sites de recherche des liaisons avec des protocoles de communication divers : X25 (de France Télécom), DECNET (de DEC), SNA (d'IBM), ISO (très fortement poussé par la communauté européenne) et quelques IP (très fortement poussé par les universités américaines). Aujourd'hui, nous ne connaissons qu'une seule langue pour les communications : IP, alias Internet. Cela n'a pas toujours été le cas, on parlait alors plusieurs langues. Pour la définition de RENATER, il avait été créé autant de groupes de travail que de protocoles de communication. L'objectif était de pouvoir utiliser tous ces protocoles sur un même réseau. Bien qu'étant jeune et sans vraiment d'expérience, j'avais été désigné comme animateur du groupe technique IP chargé de définir le modèle d'architecture (interconnexion des sites : équipements, routage...). Les autres groupes étaient pilotés par des seniors, avec des responsabilités importantes dans certains organismes de recherche. Je pense que le ministère de la Recherche et de l'enseignement supérieur et France Télécom ne croyaient pas du tout en cette langue inconnue en France d'où ma désignation.

À l'arrivée, RENATER n'a utilisé que le protocole IP. Pourquoi ? Le travail du groupe technique IP a été de très bonne qualité mais ce n'est pas ce qui a conduit à la décision de ce choix. Il me semble que deux éléments

1. <http://www.renater.fr>.

importants ont tranché. Le premier est que IP est simple, donc il fonctionne avec tous les débits de ligne, avec des équipements peu coûteux et avec des logiciels légers. Ce sont les éléments du succès d'Internet. Le second point est que IP s'est répandu rapidement dans le monde. Et, même dans nos villages gaulois, on ne peut pas ignorer le poids du plus grand nombre, du contexte international, dirait-on actuellement. Quand un standard, un logiciel s'impose mondialement, il est difficile de prendre une autre solution.

Mais quel rapport avec le logiciel libre ? Il existe une similitude sur le fond.

Pour que les logiciels libres se répandent, il faut qu'ils soient simples et légers, avec des fonctions de base : *keep it simple*. C'est l'efficacité du pragmatisme anglo-saxon face aux constructions intellectuelles grandioses de notre vieille France. Et il me semble que dans le libre, les outils légers et très ciblés ont beaucoup de succès.

La transcription du second point est qu'on ne peut pas lutter contre Microsoft Office. De même que IP est devenu la langue naturelle du réseau, le format *.doc* est le format le plus répandu dans le monde de la bureautique. On ne peut l'ignorer. Il aurait été suicidaire au niveau du CNRS de vouloir imposer OpenOffice comme première étape de la promotion du logiciel libre, ce que j'aurais pu néanmoins proposer plusieurs années après. Microsoft Office est trop répandu, au CNRS comme partout dans le monde. J'utilise personnellement LibreOffice et je recommande de l'utiliser, mais je n'ai jamais recommandé de basculer systématiquement tous les postes personnels des laboratoires sous cette bureautique libre en supplantant Microsoft Office. Le positionnement des Directions des Systèmes d'Information (DSI) centrales (voir un paragraphe plus loin) intervient aussi dans cette position pragmatique. Dans la promotion du logiciel libre, certains pensaient qu'il fallait commencer par là : le poste utilisateur sous Linux pour tous. Je ne sais pas ce qu'il en est aujourd'hui. Je pensais (et je pense toujours) le contraire, dans mon environnement professionnel.

Le Libre, un des éléments clés du succès de l'Internet et de ses applications (1992-2000)

À partir de 1992, et pendant de nombreuses années, l'objectif principal de l'UREC fut le déploiement des applications internet dans les laboratoires et les campus universitaires. Cela avec très peu de moyens humains. Un important travail de compilation, de tests puis de diffusion d'informations fut effectué. Nous avons organisé un site Web qui a été longtemps la référence pour IP et ses applications dans le monde académique, jusqu'à 500 heures de formations ciblées (courtes ou généralistes), des conférences, des groupes de travail pour tous les administrateurs systèmes et réseaux de laboratoires. Nous avons également formé des formateurs en interne pour qu'ils puissent intervenir à leur tour. Sans modestie, ceci fut très efficace.

Ainsi, la recherche et l'enseignement supérieur ont été longtemps en avance dans le domaine de l'utilisation d'Internet (ce n'est plus le cas actuellement). Je suis moi-même très souvent intervenu dans des conférences pour les informaticiens ou décideurs de sociétés privées, parfois dans des comités stratégiques de grands groupes industriels pour expliquer pour quoi on utilisait Internet et comment. Je me souviens du cri du cœur d'un DSI d'un grand groupe disant : « Je ne comprends pas pourquoi nous dépensons autant d'argent pour notre messagerie électronique alors que vous faites tout avec des logiciels gratuits. »

Effectivement, cela nous paraissait naturel, mais tout a été basé sur du logiciel libre qu'on a pu étudier, tester, utiliser sans problème pour les formations, rediffuser dans les laboratoires... Toute cette mise en place des applications internet s'est effectuée sans aucun investissement en achat de licence logicielle ! Outre l'aspect financier, la liberté offerte par le libre a permis cette efficacité de diffusion avec très peu de moyens humains.

Ainsi, il est évident que tous les logiciels libres ont contribué de manière fondamentale au succès d'Internet, et pas uniquement dans la recherche.

Regrouper les personnes qui savent et qui pratiquent pour être efficace dans le domaine de la sécurité informatique (1995-2006)

Parallèlement à mon travail sur le réseau et ses applications, de 1990 à 2006, j'ai été chargé de mission Sécurité Informatique à temps partiel auprès du fonctionnaire de Sécurité défense (FSD) du CNRS. Les premières années, j'étais le seul référent technique national sur la sécurité informatique au CNRS, un peu l'équivalent d'un Responsable de la Sécurité des Systèmes d'Information (RSSI). Le terme n'existait pas à l'époque, ni le poste au CNRS (le premier poste de RSSI n'a été créé au CNRS qu'en 2011).

Un travail urgent s'est rapidement imposé : verrouiller les accès internet. Pour les 1 300 laboratoires du CNRS de l'époque, tous connectés à l'Internet, il n'y avait aucune organisation pour la sécurité informatique, dans un environnement peu réceptif à ces problèmes et, pour la majorité, réfractaire aux règlements et aux contraintes. Ces laboratoires CNRS (on retrouvait la même situation dans les universités) avaient été connectés à Internet (plus concrètement à RENATER) lorsqu'il était encore uniquement académique et que quasiment tous les sites étaient des universités ou des laboratoires de recherche. On était membre de la même communauté de recherche, on se connaissait et on se faisait confiance. Il n'y avait donc aucun filtrage, aucune limitation d'accès à l'entrée des laboratoires et des campus, sans compter de très nombreux serveurs ou applications sans suivi des mises à jour logicielles, avec des mots de passe triviaux. Avec l'explosion d'Internet et son ouverture, il a fallu rapidement convaincre puis aider tous les laboratoires à mettre en place une architecture réseau sécurisée, des limitations d'accès avec une bonne configuration et un suivi de tous les serveurs. Il fallait également convaincre les utilisateurs et les informaticiens, habitués à une liberté complète, de se préoccuper de sécurité, et d'agir pour limiter cette liberté, cela rapidement. Ce n'était pas gagné.

Avec très peu de moyens, l'UREC a monté un réseau de correspondants laboratoire de sécurité informatique avec un niveau régional de coordination. Ce réseau a fonctionné jusqu'en 2006 et a compté jusqu'à 720 correspondants. Grâce à lui, nous avons pu réagir rapidement en cas d'intrusion, diffuser des recommandations, des avis de sécurité et des outils libres de

protection ou de détection et surtout sensibiliser et former. Un volume très conséquent de formations nationales, régionales... avec des formations de formateurs internes, a été maintenu pendant près de 10 ans.

Nous sommes alors arrivés au début des années 2000 à une situation saine concernant les risques Internet. La sécurité n'était pas sans faille (ce qui n'existe pas) mais satisfaisante du point de vue architecture, filtrage, administration des serveurs.

Outre l'utilisation de logiciels libres pour la protection ou la détection, un point important dans cette organisation fut la décision selon laquelle les correspondants sécurité seraient des administrateurs systèmes et réseaux dans les laboratoires, des personnes de terrain, proches des utilisateurs. Ils pouvaient donc les convaincre et changer leurs habitudes sans les heurter avec une attitude trop autoritaire. De plus, ils comprenaient les aspects techniques qu'il fallait mettre en place et ils pouvaient le faire.

Ce réseau était loin de faire l'unanimité à la direction du CNRS car il ne s'appuyait pas sur la hiérarchie organisationnelle de l'organisme : directeurs de laboratoire, délégations régionales, services centraux de l'organisme... Néanmoins il a prouvé son efficacité. Depuis 2006, il a été remplacé par une organisation beaucoup plus formelle qui suit l'organisation administrative du CNRS, avec des désignations officielles... donc lourde. Celle-ci s'est focalisée sur la rédaction de politiques de sécurité plus que sur les aspects architecture et protection opérationnelle. Mais les éléments techniques étant en place, il n'y avait plus urgence.

Une fois de plus : quel rapport avec les logiciels libres ? Outre l'utilisation d'outils libres de sécurité, la communauté des correspondants sécurité avait toutes les caractéristiques d'une communauté de développeurs : des personnes motivées (volontaires), compétentes techniquement et utilisatrices sur le terrain. On retrouve là les caractéristiques des contributeurs dans le développement des logiciels libres. Comme pour l'organisation sécurité, ce sont des éléments déterminants pour le succès.

Suite à l'élaboration d'une stratégie CNRS de diffusion des logiciels développés, quelques points faibles des logiciels conçus dans les laboratoires (2004-2006)

J'aborde ici une spécificité des laboratoires de recherche dans le développement de logiciels.

Autour de 2005, j'étais directeur de l'UREC et donc un membre actif du Comité d'Orientation des Moyens Informatiques (COMI) du CNRS. Ce comité était chargé d'élaborer et de mettre en œuvre la politique de l'informatique scientifique au CNRS en finançant des projets structurants. Au sein de ce COMI nous avons fait le constat que de très nombreux logiciels, souvent techniquement pointus et innovants, étaient conçus dans le cadre de recherches dans les laboratoires. Certains d'entre eux ont une valeur importante car ils mettent en œuvre des résultats de recherche récents, des algorithmes novateurs, ils résolvent les problèmes techniques de demain et n'ont pas d'équivalent commercial. Mais souvent confiés à des thésards, personnel temporaire et dont le métier n'est pas le développement de logiciels très (trop) rapidement écrits pour résoudre un problème scientifique précis, dans l'urgence d'un projet, d'un contrat, ils sont impossibles à diffuser, même libres en bêta. Nous avons décidé de commencer par financer la « finalisation » de certains logiciels de laboratoire après un petit appel à projets. Ce fut un succès malgré un budget modeste. Ensuite, nous avons proposé la création d'une unité (équivalent d'un service) spécifique pour le travail d'aide au développement, de finalisation et de diffusion de certains logiciels au CNRS. L'action s'est arrêtée là, car début 2006, la direction du CNRS a changé et le COMI a été supprimé sans être remplacé par un équivalent.

À travers cette expérience, on peut lister les points faibles qui rendent très difficile la diffusion des logiciels de laboratoire même s'ils sont innovants et opérationnels :

- Il est très difficile de choisir ce qu'il faut diffuser, ce qui a une valeur. En effet, les développements sont très nombreux, donc on ne peut en finaliser (c'est-à-dire « rendre propre ») et en diffuser qu'une très faible partie. De plus, ils touchent des domaines scientifiques très divers et pointus. Il faut des compétences scientifiques spécifiques dans chaque science pour évaluer leur pertinence. Il faut également juger

de la qualité du code. Comment trier ce qui va être intéressant, ce qui est bien développé ? Qui va faire ce tri ? Dans le projet COMI, nous ne nous étions pas encore posé la question sachant que la solution ne serait pas facile à trouver.

- Les développements sont très souvent de mauvaise qualité. Comme je l’ai déjà dit, ce sont souvent des thésards scientifiques et non des informaticiens qui les développent sans méthodologie ni conseil et dans l’urgence. Ils ne peuvent pas être bien documentés, structurés... De plus, ces personnes sont de passage et laissent leur développement en l’état à leur départ sans former quiconque pour les reprendre.
- L’ergonomie et l’interface utilisateur d’un logiciel de recherche sont loin d’être une priorité et sont souvent minimalistes. Or ces aspects sont très importants dans l’adoption d’un logiciel.
- Ce n’est pas le rôle d’un chercheur de faire du développement logiciel (sauf si c’est un chercheur en informatique). Cela n’entre pas dans les éléments qui évaluent son travail et assurent des promotions. Donc ce n’est pas prioritaire.
- Un laboratoire de recherche n’est pas organisé pour diffuser du logiciel et en assurer le support. Ce n’est pas non plus sa mission. Et le transfert de ces tâches à une société commerciale est souvent délicat car les logiciels sont très spécifiques, n’intéressent que des clients très ciblés et les sociétés informatiques classiques sont mal adaptées.

Donc, même si le CNRS, organisme public national, a un fonds de logiciels très intéressant, la mise en pratique de leur diffusion est loin d’être simple, malgré la mission nationale et d’utilité publique qui lui permettrait de diffuser en libre ce qui est développé dans ses laboratoires.

La politique gouvernementale, élément déterminant dans l’utilisation ou non du logiciel libre dans la fonction publique (2006)

Mi-2006, nous avons lancé le projet national PLUME sur les logiciels à grande majorité libres, dont le porteur est le CNRS. Le point suivant décrit le projet. Ici, je voudrais mettre en exergue un élément fondamental pour le déploiement du logiciel libre dans les administrations et les organismes publics.

Au démarrage de PLUME, il existait une structure nationale, l'Agence pour le Développement de l'Administration Électronique (ADAE) avec un volet promotion du logiciel libre dans les administrations. Je ne me souviens plus des missions exactes qu'avait cette agence ni de ses réalisations. Mais en 2006, nous leur avons présenté le projet PLUME. Ils avaient été très intéressés et nous avaient encouragés à le lancer en nous donnant des conseils et des contacts très précieux. Ce soutien avait été important pour nous car c'était une recommandation officielle d'une agence gouvernementale. La direction du CNRS avait immédiatement accepté le projet.

Mais l'ADAE a rapidement disparu ainsi que la promotion du logiciel libre dans les administrations. Cela a considérablement ralenti le projet. Sans ce soutien, il fallait régulièrement convaincre les directions du CNRS, ainsi que d'autres décideurs, de son intérêt. Nous y gagnâmes un budget et des ressources minimalistes. PLUME n'allait pas dans le sens de la politique gouvernementale concernant le logiciel.

Dans les organismes publics et les administrations, un message fort du gouvernement vers le logiciel libre est obligatoire pour vraiment faire adopter le libre, ce qui n'a pas été le cas au cours de ces dernières années, jusqu'en 2012 (depuis, l'orientation gouvernementale semble avoir changé). Le CNRS est un organisme public sous tutelle d'un ministère et en conséquence il n'a pas d'engagement vers le libre. De plus, à cause de la structure des laboratoires qui sont pour la plupart mixtes – CNRS-universités et d'autres entités –, le CNRS ne peut pas seul, imposer un choix technique comme le libre pour le logiciel. Il faut que la décision soit prise au moins au niveau de tous les organismes de recherche et des universités.

Quelques éléments intéressants révélés par la mise en place de la plate-forme PLUME (2006-2012)

Le projet PLUME « Promouvoir les Logiciels Utiles Maîtrisés et Économiques dans l'enseignement supérieur et la recherche » a consisté à mettre en place une plate-forme Web contenant des descriptifs de logiciels utilisés ou développés dans les laboratoires et les universités, ainsi que des descriptions de ressources (cours, sites Web, services...) liées aux

logiciels. Ces fiches sont rédigées par des contributeurs utilisateurs ou développeurs de logiciel dans l'enseignement supérieur et la recherche. L'ensemble est indexé par des mots-clés avec une mise à jour régulière. Le but est de proposer un catalogue à jour de logiciels, à 95 % libres, utilisés en production dans la communauté. Ils sont donc installables « sans problème ». Le catalogue recense aussi des logiciels développés par cette communauté pour les faire connaître (une fonction de valorisation au sens large). Fin août 2012, ce serveur propose plus de 1 100 fiches descriptives, avec près de 850 contributeurs et le nombre de nouvelles fiches est d'environ 4 à 5 par mois.

J'ai été le responsable de ce projet et le rédacteur en chef (je relisais tout ce qui était publié) de la plate-forme jusqu'en avril 2012. C'était un bon poste d'observation. Outre les éléments décrits ci-dessus et qui sont ré-apparus avec PLUME (pas d'appui officiel du CNRS au logiciel libre, difficulté de la diffusion en libre de logiciels de laboratoires...), j'ai essayé ci-dessous de lister quelques observations qui me paraissent intéressantes, un peu en vrac :

- PLUME n'est pas une promotion des logiciels libres mais des logiciels utiles, maîtrisés et économiques comme le dit l'acronyme. En effet, le CNRS n'étant pas engagé dans une démarche vers le libre, nous n'avons pas affiché le terme libre dans le nom du projet. Je ne le regrette pas. Il est « naturel » qu'un organisme mette en place une plate-forme pour présenter les logiciels utiles aux laboratoires, que ces derniers maîtrisent (c'est-à-dire que des personnes de la communauté connaissent bien et dont le source est disponible) et qui sont économiques (pour diminuer les dépenses). Personne ne peut remettre en cause ces objectifs et on pourrait dire que tout organisme / administration / entreprise devrait les afficher. Ces buts seraient certainement mieux acceptés que ceux qui se focalisent sur l'étiquette libre. C'est une attitude pragmatique, responsable et non doctrinale. Le succès de cette plate-forme nous a confortés dans ce sens : nous avons fait le bon cadrage. La plate-forme contient ainsi quelques descriptions de logiciels non libres, de shareware... , mais en fait très peu. Dans l'organisation de PLUME, pour faire un tri des logiciels à présenter, il est convenu que lorsqu'un contributeur propose de décrire un logiciel non libre, on regarde s'il existe un produit libre similaire et de

bonne qualité : si ce n'est pas le cas, on accepte la proposition. Sinon, on regarde de plus près et on discute avec le contributeur pour savoir pourquoi il utilise le logiciel non libre... Et nous ne nous sommes jamais fâchés avec un contributeur.

- Il y a beaucoup de compétences techniques sur les logiciels dans les laboratoires et les universités. Le nombre de contributions dans PLUME est révélateur : il y a des gens qui connaissent les logiciels dans la communauté et dans des domaines très variés. Dans ce contexte privilégié, il faut tout faire pour mutualiser ces compétences. C'est là-dessus qu'est basé PLUME, sur l'agrégation de ces compétences variées qui fait l'intérêt de la plate-forme.
- Dans un projet coopératif, il est nécessaire que les personnes qui participent en retirent quelque chose. Sur PLUME, le fait que chaque fiche soit signée permet au contributeur d'être connu, donc reconnu par ses pairs. De plus, des personnes peuvent le contacter pour travailler avec lui dans le cadre de développements logiciels, de projets, cela peut influencer dans son dossier de carrière, il peut être invité à des conférences pour présenter le logiciel qu'il a décrit... Et c'est ce qui se passe !
- Dans les choix de logiciel, il faut faire confiance aux personnes du métier. Au départ de PLUME, une tendance de l'équipe technique a été de juger elle-même de la qualité des logiciels, car en tant qu'informaticiens de métier, nous savions ! Mais nous n'étions pas utilisateurs. Nous avons vite abandonné. Ce principe de jugement était une erreur. Il faut faire confiance au choix des personnes du métier qui utilisent un logiciel dans leur travail quotidien, d'autant plus si elles sont prêtes à rédiger et à signer un document descriptif du logiciel, ce qui est un engagement public de leur part.
- Un logiciel a souvent des utilisations par des métiers, des domaines scientifiques différents. En effet, il est régulièrement apparu qu'un logiciel utilisé en physique était aussi utilisé en sciences humaines mais avec une pratique et une finalité différentes. Cela a surpris les membres de l'équipe PLUME mais a augmenté leurs connaissances en voyant comment d'autres utilisent différemment le même logiciel qu'eux. Il ne faut pas trop cloisonner le domaine des logiciels.

- Le logiciel n’est pas un objet simple. De nombreuses fiches PLUME en décrivent les licences, les droits d’auteurs, patrimoniaux. . . . Car il y a un besoin d’information et d’explications. Le logiciel n’est pas une entité simple et tout chercheur ou responsable de laboratoire qui veut le diffuser doit se poser de nombreuses questions juridiques devant lesquelles il est incompetent et souvent peu intéressé par ce sujet. Ce sont autant de freins à la diffusion. Ils se demandent : est-ce que tout cela en vaut la peine ? Surtout lorsque ce n’est pas pris en compte dans l’évaluation de leur travail.
- En présentant dans PLUME les développements des laboratoires, on a travaillé très régulièrement avec les services de valorisation du CNRS. Pour ces services, le critère de réussite et d’efficacité est basé sur la quantité d’argent rentrée dans les caisses du CNRS. Il est ainsi très difficile de faire comprendre qu’un laboratoire qui distribue un logiciel gratuitement et pour une utilisation internationale acquiert une grande image de marque et donc apporte de la valorisation. Cette image permet au laboratoire d’avoir des coopérations-contrats industriels, car sa réussite logicielle inspire confiance. De plus, j’ai souvent essayé de démontrer à ces services qu’un laboratoire pouvait gagner beaucoup plus d’argent en donnant des formations sur un logiciel libre qu’il a conçu, en faisant quelques développements spécifiques autour. . . qu’en essayant de vendre quelques licences. Le modèle économique du libre n’est pas compris et sa mise en œuvre n’est pas intégrée dans le catalogue des solutions de valorisation de la recherche.
- L’anonymat des auteurs Wikipédia, la possibilité de modifier les articles de cette encyclopédie et d’en faire commerce semblent être devenus le modèle universel pour la diffusion d’article. *A contrario*, les fiches PLUME sont signées avec le nom du laboratoire ou de l’université de l’auteur, et leur licence Creative Commons est BY NC ND¹. C’est un choix délibéré. Dans une fiche PLUME, une personne décrit un logiciel qu’elle utilise dans son travail, donne son avis, indique les problèmes qu’elle rencontre, le critique. Si la fiche n’était pas signée mais anonyme, si on ne connaissait pas l’environnement de travail de cette personne, ces informations auraient peu de valeur. De même,

1. C’est-à-dire l’obligation d’indiquer l’auteur, pas d’utilisation commerciale et pas de modification.

une fiche est un tout, avec une partie fonctionnalités, une partie utilisation, une partie critique, etc. Si elle pouvait être librement modifiée, les critiques seraient rapidement effacées par les promoteurs du logiciel, et si elle était découpée, le résultat serait de piètre valeur. D'où le choix de la clause ND (pas de modification). Quant à la clause NC (pas d'utilisation commerciale), c'est un engagement vis-à-vis des contributeurs. PLUME ne veut pas qu'un contributeur découvre son travail sur un site commercial avec son nom encadré par un ensemble de bandeaux publicitaires.

- PLUME est un producteur d'information. En effet, chaque fiche est une rédaction personnelle, et non un copier-coller d'une documentation ou de pages Web de Wikipedia ou d'ailleurs. On veille à cette ligne éditoriale lors des relectures. D'autres sites font uniquement de l'agrégation d'informations existantes. Ce sont souvent les plus visibles et ils sont cités comme autant de modèles. Pour notre vie de citoyen du monde, il est important qu'il y ait des sites de production d'information et de création. Dans le cas contraire, on retrouve les mêmes informations partout. Actuellement, par exemple, on assiste à la reproduction sans fin des mêmes dépêches d'actualités dans tous les médias d'information : quelle tristesse et quel danger pour la diversité et la liberté !
- PLUME est-il utile et a-t-il une influence sur l'utilisation des logiciels libres dans les laboratoires ? Il est difficile d'apporter une réponse ferme à base d'éléments chiffrés indiscutables. Néanmoins, le nombre d'accès n'a cessé d'augmenter depuis le début du projet et les contributions continuent. Un autre fait récent est venu concrétiser un des objectifs de PLUME. Au niveau du ministère de l'enseignement supérieur et de la recherche, un groupe d'experts négocie des tarifs de licences d'utilisation auprès des distributeurs de tous types de logiciels utilisés dans les universités et les laboratoires de recherche. Début 2012, ce groupe a contacté PLUME pour qu'un de ses membres participe à leur travail (afin d'apporter une connaissance de l'existant en libre). Sur leur site Web, ce groupe a décidé qu'à côté des licences commerciales, il y aurait des pointeurs vers les fiches PLUME décrivant les logiciels libres équivalents. C'est un objectif de PLUME : indiquer aux laboratoires les logiciels libres qu'ils peuvent utiliser à

la place des logiciels commerciaux et leur donner toutes les informations pour leur permettre de choisir. D'après ces éléments, on peut penser que PLUME a un effet positif sur l'utilisation et le développement de logiciels libres dans la recherche.

Le prix et l'accès aux documents qui décrivent les standards (2012)

Pour changer de domaine et être plus léger, en 2011-2012, j'ai été chargé de mission qualité (au sens large) à temps partiel dans un gros laboratoire. Ce domaine est très intéressant si on sait prendre du recul. J'ai fait commander un document qui décrit la norme ISO 9001, c'est-à-dire 30 à 40 pages dont la moitié en références, non relié. Coût : 78 €. La documentaliste était stupéfaite et moi aussi. Par comparaison, les RFC (*Requests for Comments* – demande de commentaires), qui sont les documents qui décrivent les standards Internet, sont des documents gratuits, disponibles en ligne (j'en ai lu attentivement certains pour comprendre et faire des cours). En sachant cela, on comprend mieux ce qui a fait le succès d'Internet : les standards et autres documents de référence sont publics et gratuits.

La DSI des organismes de recherche est souvent un frein au Libre

Pour terminer, signalons une caractéristique de l'immense majorité des Directions des Systèmes d'Information de toutes les administrations et entreprises. Au CNRS, une DSI a en charge toute l'informatique concernant la gestion administrative, financière et les ressources humaines. Comme dans la plupart des cas, ses choix techniques sont Oracle, SAP, Microsoft. Les laboratoires utilisent obligatoirement certains logiciels de gestion fournis par cette DSI. Souvent, ils ne tournent que sous Windows et imposent Internet Explorer comme unique navigateur. Donc, chaque laboratoire doit avoir au moins un équipement Windows. Cela ne facilite pas le passage aux logiciels libres.

Conclusion personnelle

Les faits racontés ci-dessus sont extraits de mon parcours professionnel. Ils sont donc divers, disparates et partiels. Mais ils ont été choisis

comme des coups de projecteurs sur l'histoire et sur des spécificités de l'environnement recherche, en lien avec le logiciel libre. Chaque fait essaie de donner un éclairage différent, d'apporter une idée différente. Ces petites histoires peuvent aider à comprendre pourquoi les laboratoires de recherche utilisent partiellement les logiciels libres et diffusent très modérément leur production logicielle. Les faits sont multiples car la réponse est un ensemble de raisons diverses. Il n'y a donc pas vraiment de conclusion à cet article.

Néanmoins, un point important n'a pas été abordé : par ses propriétés, le logiciel libre est vraiment très bien adapté à la recherche publique et à la démarche scientifique. En effet, une des missions fondamentales du CNRS est la diffusion des connaissances. Si on l'applique aux logiciels développés dans un laboratoire, cela impose de les diffuser en libre (avec le source. . .). Ensuite, concernant la démarche scientifique, il faut pouvoir vérifier toutes les expériences de manière à les reproduire. Pour l'outil informatique, cela impose d'utiliser uniquement des logiciels dont le source est accessible (pour vérifier ce qu'ils font réellement) et qui peuvent être ré-utilisés (et éventuellement adaptés) autant de fois qu'on veut pour reproduire l'expérience.

Dans les laboratoires de recherche publique, on ne devrait produire et utiliser que des logiciels libres.

Annexe

Liste des auteurs

Adrienne ALIX Directrice des programmes de Wikimedia France (Paris), s'occupant notamment de définir, coordonner et suivre les différents programmes de l'association : soutien et valorisation des projets Wikimedia dont Wikipédia, soutien à la contribution et à la libération de contenu. Contributrice sur Wikipédia depuis 2006 et historienne de formation.

Jean-Luc ARCHIMBAUD Ingénieur en informatique (CNRS) diplômé de l'INSA Lyon, actuellement responsable de projets et directeur adjoint technique de l'unité CNRS Mathdoc. Il a lancé (et en a été responsable pendant 5 ans) la plate-forme PLUME (<https://www.projet-plume.org>). Voir aussi <http://jl.archimbaud.free.fr>.

Sébastien BROCA Docteur en sociologie (Cetcopra, Université Paris 1 Panthéon-Sorbonne). Auteur d'une thèse intitulée *L'utopie du logiciel libre. La construction de projets de transformation sociale en lien avec le mouvement du free software*. (dir. Ph. Breton, 2012).

Jonathan CHIBOIS Membre du IIAC – LAIOS (UMR 8177), École des Hautes Études en Sciences Sociales, Paris. Doctorant en anthropologie politique, il prépare une thèse sur l'évolution des pratiques numériques des députés de l'Assemblée nationale française. Carnet de recherche en ligne : <http://laspic.hypotheses.org>.

Damien DJAOUTI Maître de conférences en informatique à l'IUFM de l'Université Montpellier II - laboratoire LIRDEF (<http://www.lirdef.fr>). Il travaille sur la question des *Serious Games*, ces jeux

vidéo destinés à d'autres finalités que le divertissement : éducation, communication, santé... Co-fondateur de l'association Ludoscience (<http://www.ludoscience.com>) dédiée à l'étude du jeu vidéo.

Primavera DE FILIPPI Chercheuse associée au CERSA / CNRS / Université Paris II. Experte légale à Creative Commons France et cofondatrice du chapitre français de l'Open Knowledge Foundation.

Séverine GIORDAN Docteure en sciences de l'information et de la communication. ATER à l'Université d'Avignon dans le Master stratégie du développement culturel mention « médiations de la culture et du patrimoine ». Elle s'intéresse aux espaces sociaux et culturels reposant sur des processus de gouvernance partagée. Spécialiste du *Data Curation*, elle porte son attention principalement sur la constitution des mémoires collectives des ces groupes sociaux et leurs diffusions entre espaces présents et virtuels (fictions, expositions, débats, délibérations).

Pierre-Amiel GIRAUD Membre de l'UMR 5185 ADES – CNRS / Université de Bordeaux. Agrégé de géographie, il prépare une thèse sur les lieux et les territoires de la mouvance du Libre, en optant pour une démarche comparative des pratiques aquitaines et québécoises. Il cherche notamment à y montrer qu'Internet, en tant qu'espace géographique, ne rend visible qu'une partie des phénomènes essentiels à la compréhension de la mouvance du Libre.

Gérard GIRAUDON Directeur de Recherche Inria, spécialiste de Vision par Ordinateur, actuellement directeur du Centre de Recherche Inria, Sophia Antipolis – Méditerranée. Il a été pendant une dizaine d'années en charge du Développement et des Relations Industrielles d'Inria. Il a contribué à structurer la stratégie de l'Institut sur l'*open source* et la mise en œuvre de sa politique comme outil de transfert notamment à travers la création de consortium (ObjectWeb, Scilab, etc.). Il est l'un des co-auteurs des licences de la famille CeCILL. Il a publié en 2006 une tribune dans *Les Échos*, intitulée « Le logiciel libre et ouvert : révolution ou évolution ? », dont on peut trouver une version longue à http://interstices.info/jcms/c_14658/le-logiciel-libre-et-ouvert-revolution-ou-evolution.

Patrick GUILLAUD En charge des relations entre STIC et société à la direction du développement technologique d'Inria. Il pilote le

concours *Boost your code*. Son cursus professionnel l'a mené de l'industrie à l'enseignement et la recherche. Ayant une formation en automatique, architecture de systèmes, gestion, et un doctorat en sciences de l'information et de la communication, ses recherches portent sur la créativité et l'innovation technologique. Il enseigne à l'INSA de Lyon.

Alexandre HOCQUET Université de Lorraine & EHESS, Nancy & Paris. Professeur des Universités en Chimie (<http://hocquet.eeigm.eu>), étudiant en Histoire des Sciences et des Techniques à l'EHESS, entraîneur de Football sur Internet.

Thibaud HULIN Université de Montpellier 2, maître de conférences en sciences de l'information et de la communication, Laboratoire Interdisciplinaire de Recherche en Didactique, Éducation et Formation. Philosophe de formation, développeur occasionnel de logiciel libre.

Benjamin JEAN Membre de l'équipe du CUERPI (Centre Universitaire d'Enseignement et de Recherches en Propriété Intellectuelle), cofondateur de la société Inno³, consultant au cabinet Gilles Vercken et maître de conférences à Sciences Po. Cofondateur de l'association Veni, Vidi, Libri et du cycle de conférences European Open-Source & Free Software Law Event (EOLE).

Nicolas JULLIEN Maître de conférence en économie à Télécom Bretagne. Il travaille sur les communautés en ligne, et notamment le logiciel libre depuis 1998. Il a publié une dizaine d'articles scientifiques sur le sujet.

Hervé LE CROSNIER Enseignant-chercheur à l'Université de Caen, il dispense des enseignements sur les technologies du Web (informatique) et la culture numérique (sciences de l'information). Ses recherches portent sur les relations entre l'Internet (et plus généralement le numérique) et la société. Il travaille également sur la théorie des biens communs et sur la communication scientifique. Il a publié dans la collection Les Essentiels d'Hermès deux ouvrages : *La Neutralité de l'internet* (avec V. Schafer et F. Musiani) et *La propriété intellectuelle : Géopolitique et mondialisation* (avec M. Dulong de Rosnay).

Christophe MASUTTI Docteur en histoire des sciences et des techniques, chercheur associé au SAGE / UMR 7363 (Sociétés, Acteurs, Gouvernements en Europe) à l'Université de Strasbourg, responsable des affaires européennes à la Direction Générale des Hôpitaux Universitaires de Strasbourg. Président de l'association Framasoft depuis janvier 2012.

Camille PALOQUE-BERGES Docteure en sciences de l'information et de la communication et actuellement post-doctorante LabEx HAS-TEC - DICEN (CNAM). Travaille sur l'histoire d'Internet et des pratiques de communication en réseau. Membre des associations WebArchivists.org et Patrimoine Belfortain de l'industrie informatique (PB2i). Auteure de *Poétique des codes* (Paris : Archives contemporaines, 2009).

Isabelle RAMADE Isabelle Ramade, Professeure agrégée affectée à l'université Paris-Sud 11. Chargée de mission C2i (Certificat Informatique et Internet) et Responsable Ressources Pédagogiques Numériques à la Faculté des Sciences d'Orsay. Responsable des Ressources Éducatives Ouvertes à Creative Commons France. Elle promeut la culture numérique à travers le C2i, et une culture libre à travers ses engagements personnels depuis 2005 (AFUL, Parinux, FDN et Wikimedia France), et depuis 2012 avec Creative Commons France.

Stéphane RIBAS Passionné par la collaboration en mode distribué dans le domaine du développement logiciel, il travaille à la direction du développement technologique d'Inria, où il assure la promotion des méthodes de management de communautés auprès des équipes de recherche de l'institut. Il est l'organisateur des conférences FOSSa (Free Open Source Software for Academia) et il aime faire partager son expérience du libre auprès des étudiants grenoblois.

Pascal ROBERT Professeur en sciences de l'information et de la communication à l'ENSSIB/Université de Lyon et membre du laboratoire Elico. Il a récemment publié *Mnémotechnologies, une théorie générale critique des technologies intellectuelles* (Paris : Lavoisier, 2010) et *l'Impensé informatique, critique du mode d'existence idéologique des TIC. Vol. 1. Les années 1970-1980* (Paris : Archives contemporaines, 2012).

Valérie SCHAFER Chargée de recherche, Institut des sciences de la communication du CNRS (Paris). Historienne. Auteur de *La France en réseaux (années 1960/1980)* (Paris : Nuvis, 2012) et *Le Minitel, l'enfance numérique de la France* avec Benjamin Thierry (Paris : Nuvis, 2012). Organisatrice avec Fanny Georges du séminaire *Historiques de l'Internet* à l'ISCC.

Stéphane UBÉDA Professeur des universités à l'Institut National des Sciences Appliquées de Lyon, il met sur pied et dirige pendant 10 ans un laboratoire de recherche en télécommunications et systèmes distribués. Associé à Inria comme chef d'une équipe mixte pendant cette période, il entre en 2009 au comité de direction de cet institut comme directeur scientifique en charge du développement technologique.

Robert VISEUR Assistant à la Faculté Polytechnique de l'Université de Mons, en Belgique, et Ingénieur de recherche au CETIC (Centre d'Excellence en Technologies de l'Information et de la Communication). Ingénieur civil et Docteur en sciences appliquées de formation, il travaille principalement sur la valorisation des logiciels libres (modèles d'affaires, licences, etc.) et sur la mise en œuvre des technologies libres de recherche d'information.

Jean-Benoît ZIMMERMANN Directeur de Recherche CNRS, GRE-QAM, Université d'Aix-Marseille, Aix-Marseille School of Economics. Économiste, il travaille sur l'approche des interactions comme fondements de la coordination des agents économiques. Travaux sur l'industrie informatique, le logiciel et le logiciel libre depuis 1999 (une douzaine d'articles publiés).

Table des matières

Préface	
<i>Hervé Le Crosnier</i>	v
Introduction générale	
<i>Camille Paloque-Berges et Christophe Masutti</i>	xiii
Notes à propos de l'ouvrage	xx
Remerciements	xxiii
I (Pré-) histoire	1
Des réseaux ouverts, pour quoi faire ?	
<i>Valérie Schafer</i>	3
1. Les réseaux seront ouverts ou ne seront pas	5
2. Du réseau « libertaire » au réseau « libéral »	16
Conclusion	25
Ingénieurs, hackers : naissance d'une culture	
<i>Christophe Masutti</i>	31
1. À propos d'ordinateurs et de temps partagé	32
2. Moufettes et canard sauvages	38
3. Le travail hacker	47
4. Réseau	55
5. Une contre-culture ?	61

Esquisse d'une archéologie de l'informatique communicante	
<i>Pascal Robert</i>	67
1. L'augmentation et le bootstrapping	68
2. La modélisation, la résolution de problèmes et le réseau	79
Conclusion	90
Sharing Source Code	
<i>Christopher Kety (trad. Camille Paloque-Berges)</i>	93
II Économie du Libre	97
Les modèles économiques du logiciel libre et leur évolution	
<i>Stéphane Ribas, Patrick Guillaud, Stéphane Ubéda</i>	99
1. Années 1960 et 1970, les débuts de l'informatique : des modèles économiques basés sur le matériel	103
2. Années 1980, l'époque des pionniers du Libre	105
3. Années 1990, naissance de GNU/Linux : du logiciel libre à l'Open Source Initiative (OSI)	108
4. Première décennie 2000, professionnalisation de l' <i>open source</i> et structuration d'un nouveau secteur économique	114
Conclusion	131
Le logiciel libre : un renouveau du modèle industriel coopératif de l'informatique	
<i>Nicolas Jullien et Jean-Benoît Zimmermann</i>	135
1. Le modèle coopératif de production du logiciel, des origines au renouveau porté par Internet	138
2. Les modèles industriels du Libre	147
Conclusion	162
Les RMLL, haut lieu mobile du libre francophone	
<i>Pierre-Amiel Giraud</i>	165
1. Les RMLL, de l'ancrage aquitain au lieu mobile	167
2. Un haut lieu du Libre, point d'entrée de la mouvance mondiale sur les territoires	180
3. Conclusion : les RMLL, lieu privilégié d'observation de la mouvance	190

III Trajectoires du Libre 195

Influence du Libre dans l'histoire du jeu vidéo

Damien Djaouti 197

1. Le Libre comme philosophie originelle des jeux vidéo ? 198
2. Du shareware au modding 207
3. La création amateur de jeu vidéo : de l'*open source* au Jeu
2.0 214
- Conclusion 221

Brève histoire de l'identité visuelle du Libre

Thibaud Hulin 227

1. Les quatre registres de libertés et de contraintes 228
2. L'énonciation éditoriale dans la mythologie Mozilla 231
3. Y a-t-il une ergonomie propre au libre ? Le cas KDE 237
4. OpenOffice et l'apprentissage de la grammaire visuelle 239
5. Comment Ubuntu ordonne-t-il le « fatras sémiotique » de
ses utilisateurs 243

Free software et open source : utopie et idéologie

Sébastien Broca 253

1. Le mariage de raison entre GNU et Linux 254
2. La naissance du mouvement *open source* 257
3. Le pragmatisme contre l'idéologie 262
4. L'idéologue n'est pas nécessairement celui qu'on croit 267
5. Un débat obsolète ? 269

IV Licences libres 275

L'évolution des licences libres et *open source* : critères, finalités et complétude ?

Benjamin Jean 277

1. La distribution des cartes : le processus d'apparition des
licences libres 281
2. La consolidation et l'harmonisation des outils juridiques
(depuis 2007) 309
- Conclusion 335

Libre choix ou choix du Libre ?***Primavera De Filippi et Isabelle Ramade* 341**

- 1. Genèse des licences libres 342
- 2. Du libre à l'ouvert ? 349
- 3. Une question de liberté 365
- Perspectives et conclusion 374

V Études de cas 379**Mozilla : histoire d'une libération*****Robert Viseur* 381**

- 1. 1993 – 1997 : héritage de Mosaïc et percée de Netscape 383
- 2. 1998 – 2002 : libération de Netscape et victoire de Microsoft 385
- 3. 2003 – 2007 : fin de Netscape et succès de Firefox 389
- 4. 2008 – 2012 : la concurrence de Webkit 395
- 5. Discussion 398
- Conclusion 401

Pure data*, logiciel libre de productions créatives**Séverine Giordan* 407**

- 1. *Pure Data*, un logiciel de programmation visuelle 408
- 2. Dialogue entre scientifiques et artistes 409
- 3. Questions de licences 413
- 4. La Communauté 416
- 5. Les atouts de *Pure Data* 423
- Conclusion 426

Wikipédia et le monde du Libre***Adrienne Alix* 431**

1. De Nupedia et GNUpedia à Wikipédia, un projet né sur les bases du logiciel libre 433
2. Les Wikipédiens, une communauté issue du monde du Libre 437
3. Des règles de vie au sein d'un des plus grands projets collaboratifs du monde 446
4. Wikipédia, un projet libre au-delà du Libre 450
5. Wikipédia, un projet politique ? 454
- Conclusion 459

Du logiciel libre pour l'Assemblée Nationale***Jonathan Chibois* 461**

1. Genèse 463
2. Réalisations 468
3. Confrontations 472
4. Usages 477
- En conclusion 482

RTFM! La chimie computationnelle : une communauté scientifique loin du libre***Alexandre Hocquet* 487**

1. RTFM! What FM? 487
2. La chimie computationnelle 489
3. Le terminal graphique 492
4. Cinq pratiques caractéristiques du Libre 494
5. Trois contraintes à la pérennité du Libre 496
6. La rupture du *desktop modeling* 498
7. La « généralification » des logiciels 503
8. What FM? Tensions dans la communauté 506

VI	Témoignages : Libre et institutions	511
	Un point d'histoire de la stratégie d'Inria sur l'<i>open source</i>	
	<i>Gérard Giraudon</i>	513
	La place du logiciel libre au CNRS	
	<i>Jean-Luc Archimbaud</i>	523
	Quelques faits passés	525
	Conclusion personnelle	540
	Annexe	545
	Liste des auteurs	545
	Table des matières	556