

Ingénieurs, hackers: naissance d'une culture

Christophe Masutti

Version 1.1



Article paru dans : Camille PALOQUE-BERGES & Christophe MASUTTI (dir.), *Histoires et Cultures du Libre. Des logiciels partagés aux licences échangées*, Paris/Lyon : Framasoft, Coll. Framabook, mai 2013, pp. 31–65.



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*, qu'il intitule « Qu'est-ce qu'un ingénieur doit savoir à propos de la programmation informatique ? »². 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.

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>.

2. 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).

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.

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 LISP³, 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 travaille sur ce système à un certain niveau n'a pas de visibilité sur

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

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.⁴

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’accès, 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

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

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 Pfaffenberger⁵ 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 »⁶. 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

5. 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.

6. 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>.

ensemble de valeurs de la sphère informatique à la sphère sociale : le partage de l'information, l'esprit d'innovation (d'amélioration), 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 Moufettes et canard sauvages

À partir de la fin de la Seconde Guerre mondiale, on voit apparaître dans l'industrie aéronautique américaine l'expression *skunk works* (littéralement, travaux de moufette). 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 investissement⁷. Les *skunk 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 majeure⁸, la Lockheed Corporation se devait de conserver secrets certains développements quitte à ne pas élaborer de stratégie claire pouvant révéler des intentions sur la création de prototypes. Les *skunk 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).

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

8. *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 ces téléphones sonnaient. Les services secrets américains avaient régulièrement intercepté des références aux skunk 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. »

Les *skunk works* sont donc des dispositifs d'organisation en ingénierie⁹ 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 *skunk 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 soustraire 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 *skunk 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 *skunk 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 dé-

9. Shane Greenstein, spécialiste en économie de l'innovation en informatique, montre comment les *skunk 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>.

crits Steven Levy¹⁰ 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, 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 Hollerith¹¹. En cela, cette fusion préfigurait ce qui allait devenir une véritable culture d'entreprise chez IBM : rassembler les nouveautés et assembler les compétences¹². 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 secrets¹³, 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 *skunk works*, cependant, l'objectif voulu et décrit par T. Watson Jr.¹⁴ consistait à encadrer les travaux de groupes auto-formés, par affinité, identifiés parmi la population des ingénieurs-programmeurs 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

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

11. 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.

12. 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>.

13. 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.

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

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, il 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 canard 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.¹⁵

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 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 mémoire¹⁶.

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*

15. *ibid.*, p. 14.

16. 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.

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).

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'attendent 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'exemplaires¹⁷. 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)¹⁸.

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

17. 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.

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

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 interviews¹⁹. 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 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

19. 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.

sur les principes managériaux de l'industrie (*wild ducks*) mais de manière autogérée : 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 hacker²⁰ :

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*²¹. 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 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*).²²

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 Diamond²³, Torvalds se décrit comme un hacker, nonobstant le

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

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

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

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

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 adhérer²⁴. 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 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*²⁵.

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 améliorations²⁶. 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

24. 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/linus-torvalds-l-interview-anniversaire-des-20-ans-du-noyau>.

25. 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.

26. 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.

les hackers²⁷.

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 1975²⁸, 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".*

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

27. 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'exécuter 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.

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

(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. Tirole²⁹, 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. Hermalin³⁰ sur le leadership, ils se réfèrent eux aussi à Max Weber et sa sociologie³¹ 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 facteurs 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 a 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*³². 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.³³

L'appellation de hacker possède donc plusieurs acceptions et il est difficile de déterminer avec exactitude comment une communauté de hackers a 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

29. 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).

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

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

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

33. 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.

et de l'utilisation.

En effet, à la différence des journalistes écrivant les épisodes biographiques qui marquèrent 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 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.³⁴

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'acceptation 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 technologies³⁵ 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 /

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

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

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. »³⁶

Longtemps considérée comme l'incubateur principal d'un « projet Internet », la DARPA³⁷ 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 ouvrages³⁸ 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

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

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

38. 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.

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 organise la circulation de l'information de point à point par un seul chemin élaboré au besoin (« Mademoiselle, passez-moi le 22 à Anières »³⁹).

Faire circuler l'information par paquets⁴⁰ 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 militaire⁴¹. 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*⁴². 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 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 Greenstein⁴³, 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,

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

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

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

42. 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>.

43. GREENSTEIN, *op. cit.*

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 Serres⁴⁴, 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'affinités⁴⁵ : non seulement un nombre important de chercheurs et d'étudiants devinrent familiers avec les principes développés par l'ARPA, mais c'est 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 Linux⁴⁶, 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

44. 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.

45. 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.

46. 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>.

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 « universitaro-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 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 informatifs⁴⁷... 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 prémises 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 calcul⁴⁸. En somme : une optimisation des ressources du système et du temps de calcul disponible, pour un système informatif permettant de gérer les tâches de chacun.

Il faut lire entre les lignes de l'introduction du manuel de l'ITS⁴⁹, 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

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

48. 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 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.

49. [ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-161A.pdf](http://publications.ai.mit.edu/ai-publications/pdf/AIM-161A.pdf).

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 Roszak⁵⁰ analysant les 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 »⁵¹, 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,

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

51. 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.

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 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), p. 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 Philipp 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'ARPANET », 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.