

Internet et le logiciel libre



our y voir clair, il n'est pas inutile de se souvenir que l'informatique dans son ensemble n'a pas été conçue comme un instrument de libération mais comme un instrument pour garder et accroître le pouvoir. L'avènement de la micro-informatique à la fin des années 70 pouvait faire espérer un rééquilibrage des pouvoirs au niveau du matériel. En France, cet espoir était nourri par l'initiative de quelques informaticiens réunis autour de Jean-Pierre Bouhot qui ont présenté au Sicob 78 un « ordinateur dans un sac ». Rapidement, cet espoir fut balayé quand l'industrie informatique de gros calibre (IBM, INTEL) eut compris qu'il était possible de vendre plus cher des matériels de moindres performances.

Mais le matériel seul ne suffit pas : il faut que la culture suive, il faut que les utilisateurs prennent conscience des moyens dont ils

disposent, qu'ils apprennent à s'en servir pour leurs besoins réels et non pas (ou pas seulement - si on reste modeste) pour les besoins inoculés par voie publicitaire ou imposés parce que c'est le bon plaisir de l'informaticien plénipotentiaire. L'avènement des logiciels dits « libres » permettra-t-il de résister et peut-être même de regagner du terrain perdu ces vingt dernières années ?

Il se peut que le terme « libre » pour qualifier un mode de diffusion de logiciels soit mal choisi, mais il faut voir l'articulation du terme : il s'oppose aux liens et dépendances qui résultent du mode de diffusion du logiciel propriétaire.¹

L'informatique : science et technique

L'informatique est un multiplicateur de mémoire : une copie est vite faite. Couplé au réseau téléphonique mondial, elle est aussi un accélérateur de mémoire : ce que j'écris en ce moment peut quasi instantanément être lu en n'importe quel point du globe - à condition que quelqu'un le lise et pas seulement des moteurs d'indexation. Pour que le texte soit effectivement trouvé et lu, il faut un certain nombre de conditions favorables. Créer ces conditions est une tâche complexe et ce n'est pas idiot de dire que c'est une tâche stupide : au lieu de réfléchir au problème, il faut réfléchir à sa mise en scène. On peut éventuellement faire confiance à un moteur de recherche quand il ne trouve rien pour un mot-clé donné ; quand la recherche est fructueuse, le dépouillement du scrutin est sisyphien.

Cette disponibilité, pourtant, est loin d'être « libre » : les canaux de communication (lignes téléphoniques, etc.) sont des outils propriétaires ou tendent à le devenir en ces temps de démolition des services publics. La privatisation des

canaux de l'information va de pair avec la privatisation de l'information elle-même, et non seulement de l'information, mais aussi du savoir concernant l'existant et finalement de l'existant tout court : le sommet du Pariou, une certaine molécule chimique fabriquée par une plante, le gène du tour de poitrine, une couleur. Ce ne sont pourtant pas là des savoirs nouvellement acquis, comme par exemple une technique de synthèse chimique, mais des savoirs latents qui sont constitutifs de la planète et que personne ne peut s'enorgueillir d'avoir créés. Tout ce qui n'est pas expressément du domaine public est devenu brevetable. La distinction entre « invention » - en principe brevetable - et « découverte » est malaisée. Par exemple, en mathématiques, les outils de base (la démonstration par l'absurde, la démonstration récursive, etc.) sont connus depuis longtemps : ce sont des techniques inventées. Leur mise en application permet de défricher l'immense domaine des mathématiques. La question est loin d'être tranchée : les mathématiques existent-elles à notre insu (c'est-à-dire qu'un nouveau théorème est une découverte) ou sont-elles créées par un nouveau théorème ? Qu'en est-il des conjectures ? Qui crée davantage : celui qui pense que... ou celui qui, beaucoup plus tard, le démontre ?

Ceci pour dire que la science et le savoir progressent par la coopération. Les informaticiens du logiciel coopératif prennent acte que l'informatique est une science et un savoir. Ils préconisent donc de garantir un libre accès au savoir informatique et mathématique. En contrepartie, cette utilisation n'entraîne aucune autre entrave que l'obligation de faire revenir dans le pot commun tout nouveau développement. Les logiciels libres sont des logiciels coopératifs, au même titre que des méthodes de composition

1. « Propriétaire » est aussi un adjectif.

musicale ou les méthodes pour se gratter quand ça démange.

Dans un texte publié en 1994, *Why Software Should Not Have Owners*², Richard Stallman montre que les arguments des tenants des logiciels propriétaires entravent le développement de l'informatique et sont nocifs pour la société humaine (en tant qu'informaticien, il est en effet convaincu que ce développement est un bienfait). Transposons le plaidoyer de Stallman en un contexte moins marqué par l'idéologie américaine : le logiciel coopératif est la base constituante d'un logiciel de service public. La brevetabilité des savoirs élémentaires associée aux méthodes judiciaires selon les lois états-uniennes – selon lesquelles il suffit d'être solvable plus longtemps que l'adversaire pour avoir le droit de rafler la mise – sera ou est déjà un frein efficace au développement global, oui global, des sciences algorithmiques et mathématiques. À la limite, l'ensemble des logiciels libres ne pourrait survivre que dans une sorte de clandestinité, mais quand on sait que la prolifération de ce savoir est intimement liée au réseau internet, c'est-à-dire à une publicité mondiale, cet ensemble pourrait disparaître. Les possesseurs des canaux d'information arriveraient ainsi à une bien meilleure maîtrise des contenus de ces canaux.

Propriétaire/coopératif

Un programme existe en général sous deux formes : un texte (dit aussi texte-source ou code-source), constitué d'une suite d'instructions, rédigé par le programmeur et le programme proprement dit qui est la forme exécutable par l'ordinateur. Un compilateur ou traducteur effectue la traduction du texte-source en forme exécutable, la seule forme qu'un ordinateur comprenne : c'est une suite de

0 et de 1. La vente d'un logiciel peut donc se faire sous deux formes : soit on vend les textes-sources avec les compilateurs-traducteurs correspondants (dans ce cas, l'utilisateur peut lui-même adapter le logiciel à l'évolution de ses besoins), soit on vend l'exécutable (dans ce cas, l'utilisateur n'a plus qu'une marge de manœuvre réduite). Ce dernier cas est le mode de commercialisation dominant, c'est celui du logiciel propriétaire.

Tant que tout va bien, le logiciel propriétaire répond parfaitement aux besoins des utilisateurs. L'utilisateur disposera d'une licence d'utilisation, mais il n'aura pas accès au texte-source. Il ne pourra donc pas lui-même améliorer ou adapter le programme, mais devra passer par le concepteur qui intégrera les bonnes idées de l'utilisateur dans son produit tout en lui faisant payer la mise à jour. Dans la majorité des cas, la finalité d'un programme est de produire quelque chose : un texte, une liste comptable, un dessin, une musique, etc. Cette chose que l'utilisateur a produite est stockée selon un certain format sur son ordinateur. Dans la plupart des cas, ce format est tenu secret. Il s'ensuit que l'utilisateur ne peut accéder à son propre texte que par l'entremise du logiciel approprié. Il est littéralement ligoté à son fournisseur de logiciels. Si beaucoup de formats sont compatibles, c'est qu'ils font l'objet d'ententes entre usines logicielles dont les intérêts commerciaux sont liés. Cette compatibilité est le résultat d'ententes – ou d'armistices – entre grands et puissants. Elle n'est qu'un moyen de plus pour ligoter l'utilisateur – et collatéralement³ tenir à distance des concurrents indésirables. Les monopoles industriels tentent aussi d'imposer des protocoles

2. Pourquoi le logiciel ne doit pas être propriétaire, en général joint au logiciel emacs.

3. Oui, c'est ça.

tenus secrets pour la transmission de documents : quand le savoir-faire va plus vite et ailleurs que les prévisions des marketteurs, ce sont les protocoles issus de l'informatique coopérative qui s'imposent. Il en fut ainsi pour le protocole TCP/IP de transmission par Internet.

Retenons que dans le monde du logiciel coopératif, l'ensemble des textes-sources est ouvert à chacun et que chacun peut participer à son amélioration. Il s'ensuit que des erreurs ou de mauvaises directions de développement y sont plus vite dénichées que dans une entreprise hiérarchisée où c'est le chef de meute qui souvent emporte la décision. Il ne faut pas croire que Linux ne connaît pas ses petits chefs, mais il est plus facile de contourner leur nocivité : il suffit de les ignorer de sorte qu'ils n'aient plus prise sur le projet.

D'un point de vue technique pur, les systèmes d'exploitation libres sont bâtis sur des bases plus saines : c'est un vrai système multi-tâches multi-utilisateurs ; multi-utilisateurs parce que Linux peut gérer les particularités d'un grand nombre d'utilisateurs et multi-tâches parce qu'un programme peut être appelé simultanément un grand nombre de fois sans que Linux se mélange les crayons ou effectue des effacements intempestifs – c'est-à-dire que le plantage est encore le moindre mal.⁴

Multi-tâches et multi-utilisateurs, un système libre est actuellement le meilleur choix chaque fois qu'un grand nombre d'utilisateurs est interconnecté : instituts et laboratoires scientifiques, serveurs de réseaux, administrations, entreprises. Paradoxalement, la diffusion réelle de Linux est inégale : les scientifiques l'ont adopté, les administrations s'y mettent car il est plus facile de mettre les données



à l'abri des regards indiscrets quand on peut consulter le texte-source des programmes. C'est illusoire quand les logiciels de cryptage de données sont eux-mêmes opaques. Bien que les logiciels coopératifs destinés à la bureautique de base soient presque gratuits, l'industrie et l'artisanat ignorent Linux pour des raisons multiples qui sont – à mon avis – conjoncturelles : des arnaques grotesques comme celle du « bug de l'an 2000 » devraient inciter à la réflexion ! Et amener un changement de cap.

Dans son ouvrage *le Culte de l'Internet*, Philippe Breton, sociologue, distingue trois populations : les partisans exaltés du tout-Internet, les technophobes et les pragmatiques. La distinction et l'analyse manquent de finesse. Aux exaltés, Ph. Breton attribue une nouvelle religiosité dont la « noosphère » constituerait à la fois le texte-tissu fondateur en retissage continu et une hyper-conscience dans laquelle pourraient se fondre toutes les consciences individuelles. Il omet de dévoiler les motivations de ces nouveaux papes (Nicholas Negroponte, Bill Gates, et quelques autres) : faire croire qu'on peut participer à une grande aventure, anesthésier autant que faire se peut. Le carburant des moteurs Gates & Cy, c'est leur soif de pouvoir et de prestige. Les suiveurs quelque peu illuminés ne sont pas moins nocifs pour autant : leur message n'est qu'une « imposture intellectuelle » parmi d'autres.⁵

4. Je parle d'expériences vécues avec d'autres systèmes.

5. Alan Sokal et Jean Bricmont, *Impostures intellectuelles*, Odile Jacob, 1997.

Thémistifications

L'avènement du logiciel libre ne peut être comprise que si on le resitue dans le contexte juridique états-unien. Le « *copyright* » n'accorde aucun droit à l'auteur-individu mais tous les droits à l'éditeur. Ce que Tzvetan Todorov dit des sciences humaines vaut aussi pour le reste : seuls les individus pensent, or sans pensée la recherche est mise à mort. Spoliés du droit d'auteur, les informaticiens-créateurs ont trouvé la solution du logiciel libre pour entamer le pouvoir des éditeurs. Mais les licences d'utilisation des logiciels libres n'interdisent pas l'assemblage, la transformation, la vente des logiciels : la seule obligation résiduelle est de fournir aussi le code-source. Cet aspect des licences du logiciel libre a un aspect kamikaze et ne résout pas les problèmes de survie du créateur-informaticien.

La teneur des licences n'exclut pas les comportements prédateurs des assembleurs (SuSE, RedHat, Mandrake) qui peuvent librement inclure dans leurs paquets logiciels des programmes dont la création et le développement a été fait par des bénévoles et ce sans aucune obligation de rémunération.

Ces fausses notes ne doivent pas faire oublier que le logiciel libre est en plein essor, et ce sont justement les prédatations effectuées par les grandes institutions qui le démontrent. Ainsi l'IRCAM (Institut de recherche et de coordination acoustique/musique) après avoir été le promoteur des systèmes propriétaires depuis toujours vient de tourner casaque et de proposer un paquetage estampillé qualité IRCAM composé de logiciels libres de musique. En effet, il est difficile de trouver dans le commerce un aussi bon logiciel de notation que la Common Music Notation de Bill Schottstaedt avec une aide par courrier électronique aux utilisateurs aussi efficace. Et ce n'est qu'un

exemple parmi d'autres. Le modèle IRCAM de la création et diffusion de logiciels a donc été mis à mal. On comprend que les grands de l'industrie informatique veulent préserver leur avenir en instaurant mondialement un droit de brevetabilité des algorithmes (parallèlement, il est clair que le parc génétique existant n'est rien d'autre qu'un ensemble d'algorithmes chimiques que l'industrie biochimique se veut approprier).

Un modèle ?

Évidemment, le logiciel coopératif peut être considéré comme un modèle de coopération applicable à d'autres domaines de l'activité. D'abord, il faudrait remarquer que la coopération est un modèle en soi s'opposant à un modèle prônant la non-coopération. Si Linux est une application réussie de la coopération, cela tient à certaines conditions qui ne se retrouvent pas forcément en d'autres domaines :

– **L'évaluation**, ou ce qu'apporte le logiciel à l'utilisateur ; cette évaluation est faite dans un délai très bref par quantité d'utilisateurs fédérés par les listes de discussion sur Internet où fusent les critiques et les suggestions.

– **L'effet cliquet** : une amélioration obtenue n'est en général pas rétrogradée, car les apports nouveaux venant des listes d'utilisateurs font rarement chavirer le projet.

– **La clarté algorithmique** : comme les textes-sources sont ouverts, chacun peut en principe suivre le cheminement de l'algorithme et intervenir là où ça fait mal.

Il n'est pas absurde de chercher à transposer ce mode d'élaboration de l'informatique théorique à d'autres domaines ; mais il y a quelques obstacles de taille.

Les autres domaines n'ont pas la même transparence mathématique : un programme informatique peut être corrigé, amélioré ou même rejeté en quelques instants (en réalité, cela ne va pas si vite, mais cela est théoriquement vrai : l'informatique, en dehors de la quincaillerie, n'est que du texte). Il n'en va pas de même quand il s'agit de transposer une idée dans un domaine à interférences humaines. Autre remarque : les acteurs de l'informatique coopérative ont souvent mis en avant la structure libertaire de leur coopération. Cela n'a pas échappé à bon nombre de « décideurs » : cohérents avec eux-mêmes, ils se trouvent des arguments pour ne pas utiliser les meilleurs outils et pour imposer des outils de moindre qualité à leur entreprise ou à leur collectivité. L'informatique n'est pas aussi désincarnée que ne le souhaitent les journalistes en mal de simplification médiatique : les décisions sont loin d'être binaires, et les choix ont bien souvent d'autres motifs que la simple efficacité. On connaît beaucoup de projets se terminant dans une impasse, car leur point de départ est idéologiquement ancré dans l'immédiat, ce qui obère toute évolution. Par exemple, on trouve beaucoup de projets informatiques de notation musicale, rares sont les projets qui sont capables de tenir compte des exigences futures des compositeurs. Plus étonnant : cela va faire bientôt cinquante ans que Noam Chomsky a étudié les langages naturels et artificiels ; on sait que la méthode la plus efficace est d'exprimer d'abord la structure d'un logiciel sous forme d'une grammaire formelle cohérente. Et que l'on parvienne ainsi à un programme stable. Si nombre d'informaticiens rechignent à se plier à cette discipline pas plus déshonorante que d'accepter que $2 + 2 = 4$, est-ce la personnalité controversée de Chomsky qui les inhibe ? Où un besoin d'obscurité ?

Les handicaps

Plus grave : il existe une certaine connivence entre les grandes industries logicielles et les grands constructeurs de matériel : les caractéristiques techniques des matériels restent secrètes de sorte que les développeurs du logiciel coopératif ne peuvent écrire les programmes permettant d'interfacer les nouveaux appareils avec le système d'exploitation Linux que lorsque ces appareils sont déjà disponibles et ce qu'après des tests et des auscultations qui prennent du temps, du temps pris sur le temps libre de ces développeurs.

Choisir un logiciel coopératif implique de s'informer préalablement et dans un organisme indépendant (association ou groupe d'utilisateurs) de ce qui est nécessaire et de ce qui est possible. C'est à chaque utilisateur de choisir le meilleur compromis – car même avec les objets, il faut d'abord s'informer, puis négocier un *modus vivendi*. Un handicap réconfortant est par contre que les matériels dont s'occupent les développeurs sont en général ceux qui connaissent la plus grande diffusion ou ceux qui sont de meilleure qualité, donc bien souvent d'un coût plus élevé. Ce qui n'est pas un obstacle : pourquoi vouloir à tout prix un ordinateur ?

Le logiciel libre est dans la continuation des mouvements d'ouverture des savoirs qui vont de Lucrèce à Chomsky. L'ambition de tout logiciel publié sous une des licences libres est de rendre la connaissance et l'utilisation la plus facile possible. Évidemment, dans la vue d'un informaticien cela veut avant tout dire : voilà le code-source. Que beaucoup de documentations soient maladroitement n'enlève rien à leur mérite, quand la composante monopolistique de l'industrie du logiciel propriétaire cherche à duper l'utilisateur. Le logiciel libre accepte toutes les collaborations : traduc-



anarca-bolo

Un sito di comunità anarchiche

teurs, correcteurs, metteurs en forme. Il est une chance de coopération entre personnes de tous bords et de toutes compétences.

La créativité informatique

L'informatique constitue un environnement mental et, en dehors de quelques détails non sans importance mais négligeables pour ce qui est de la créativité, l'évolution des techniques logicielles est essentiellement mentale. Il semble à première vue que ce sont les propositions émanant des bureaux d'études propriétaires qui façonnent cette informatique. En fait, c'est l'inverse : c'est l'utilisateur qui est le vrai créateur ; les bureaux d'études sont les tâcherons qui transposent, avec plus ou moins de bonheur, la demande de l'utilisateur en application informatique. Le but des campagnes publicitaires est tout autant de lancer le nouveau produit, l'amélioration, que de persuader l'utilisateur que c'est le bureau d'études qui a créé le produit.

Examinons d'abord le cas du logiciel

que l'utilisateur cherche. Si l'industrie du logiciel propose un outil que l'utilisateur cherche et compte tenu de ce que le logiciel est essentiellement de nature mentale, il est certain que le fait de « chercher » l'outil n'est ni plus ni moins créateur que la rédaction définitive du code-source. Il est patent dans ce cas de figure que la mise sous statut propriétaire d'un tel outil constitue un acte assimilable à un vol ou à un détournement d'un bien commun. Le logiciel comme outil mental diffère énormément d'une œuvre artistique : personne n'a éprouvé le manque d'une sérénade de Mozart avant qu'elle ne soit composée. On ne peut être en manque d'œuvre comme on peut être en manque d'outil. Les industries du logiciel exhortent les utilisateurs à faire connaître les améliorations souhaitées. Cela prouve que c'est bien l'utilisateur qui est le véritable créateur du logiciel en étant le découvreur du manque. Remarquons au passage que les jeux informatiques constituent bien davantage une création – et aussi souvent un plagiat –

qu'un logiciel de comptabilité ou de mise en page qui ne fait que reproduire informatiquement ce que la réalité exige et remarquons aussi que le statut du créateur de jeux informatiques est loin d'être clarifié : quand un individu crée vraiment, il est immédiatement en conflit avec les structures économiques.

Il en est tout autrement dans un contexte coopératif : l'inventivité personnelle bénéficie de celle des autres et elle apporte sa contribution en retour par un renvoi d'ascenseur justifié. La création artistique a un autre statut. L'œuvre d'art n'est pas perfectible ; elle est telle que la veut – ou peut – son auteur : à prendre ou à laisser. Les compositeurs ont toujours été à la fois inventeurs et auteurs : Schoenberg a inventé la méthode de composition avec douze tons. Cette méthode n'a jamais été propriétaire et a été utilisée par toute une génération de

compositeurs. Xenakis et Cage ont découvert ou inventé plusieurs généralisations de la méthode de Schoenberg, utilisables par tout un chacun, sans que cela puisse donner lieu à des procès en paternité.

Une fausse note pour finir. Le milieu hésitant le plus à s'investir dans le monde du logiciel libre est celui qui s'auto-déclare l'ardent défenseur des libertés, mais les outils dont il se sert sont majoritairement des logiciels propriétaires. Cet article a été écrit sur un système Linux avec l'éditeur emacs. Malheureusement, il sera mis en forme finale avec des logiciels du genre BackPage ou Shit Express...

René Bastian, musicien, est actif sur la scène de la musique expérimentale, en Allemagne le plus souvent. Il anime le site www.musiques-rb.org



Internet n'est pas seulement un support d'informations, il est aussi fait de logiciels qui permettent la diffusion. Dès le début les protocoles de fonctionnement du réseau ont échappé aux grandes entreprises capitalistes. L'apparition de logiciels dit libres en est la suite logique. Selon René Bastian ils sont en pleine expansion.

The Internet is not only a store of data, it is also built out of software enabling to share those data. Since its inception, the Net's protocols have stayed beyond the grasp of giant corporations. It was logical that after them open source software should be created. René Bastian considers that it has a bright future.