Preprint : Terminal N°80/81 Spécial « Les logiciels libres, de l'utopie au marché » L'Harmattan 1999 p.167-188.

Les Logiciels Libres : un nouveau modèle ?

Christian GENTHON & Denis PHAN

Faculté des Sciences Economiques, UPMF, Grenoble & ENST-Bretagne / ICI (UBO)

Résumé

Les logiciels dits "libres" suscitent actuellement un intérêt certain. Les sources de ce succès peuvent être rattachées en particulier aux caractéristiques économiques du logiciel, d'une part, et à l'évolution structurelle de l'industrie informatique, d'autre part.

Du point de vue économique, le logiciel possède certains des attributs qui caractérisent les "biens publics" (non exclusivité, externalités positives). Ces caractéristiques font que la production non marchande de logiciel a toujours existé. Dans ce monde, les pratiques développées autour de la distribution " Berkeley " du système d'exploitation Unix ont joué un rôle important dans l'émergence du mouvement des logiciels " libres ", qui doit également être interprété comme une réaction contre les pratiques de l'industrie marchande du logiciel. Quelles sont cependant les raisons qui ont permis à cette nouvelle approche de sortir avec succès de la marginalité dans laquelle elle était jusqu'alors cantonnée?

Une comparaison rapide des modes de production du logiciel libre (coopération distribuée, interaction programmeurs - utilisateurs, recours à Internet) et du logiciel marchand (centralisation, méthodologie) ne montre pas d'avantage déterminant en faveur d'un mode ou d'un autre. Le décollage du logiciel libre a été permis par l'expansion d'Internet, qu'il a favorisé et dont il a profité. La percée actuelle, qui voit le logiciel libre sortir de la niche de marché des serveurs de l'Internet, est favorisé par les stratégies des grands groupes de l'industrie informatique face à la domination actuelle de Microsoft. Tous les autres acteurs sont à la recherche d'une alternative. Le logiciel " libre ", car il suit une autre logique peut susciter le consensus de firmes en concurrence. Il se trouve donc aujourd'hui partie prenante du mouvement d'évolution de l'industrie marchande du logiciel.

Introduction

Les logiciels dits "libres" suscitent actuellement un intérêt certain. Leaders d'une niche de marché, comme le serveur Web *Apache*, ou challenger des systèmes d'exploitation de Microsoft (Windows 95 et 98, WindowsNT) sur le marché des PC comme *Linux*, leurs codes sources sont disponibles, modifiables et redistribuables, conformément à des licences spécifiques, comme par exemple la Licence Publique Générale (GPL) du GNU[1]. De telles licences ont pour fonction de les protéger contre toute tentative d'appropriation privative. La notoriété toute récente dont bénéficient les logiciels libres ne doit pas masquer les sources historiques et structurelles de ce mouvement qui peuvent être rattachées en particulier aux caractéristiques économiques du logiciel, d'une part, et à l'évolution structurelle de l'industrie informatique, d'autre part.

[1] Acronyme récursif : "*GNU's Not Unix*" associé au projet de logiciels libres de la *Free Software Foundation*.

Les spécificités du logiciel : des attributs de "bien public" et une complémentarité avec le marché des équipements (hardware)

Par certains côtés, bien que la production de code source pour logiciel se différencie de l'innovation, un logiciel est également dans une certaine mesure une "production de connaissances" et comme tel, concerné par le débat ouvert par Arrow [1962] [2]. Du point de vue économique, le logiciel possède en particulier certains des attributs qui caractérisent les "biens publics".

D'abord, l'usage d'un logiciel par un individu n'exclut pas d'autres utilisateurs, qui peuvent faire également usage de ce même logiciel en procédant à une duplication de son code (attribut de non exclusivité).

Ensuite, dans de nombreux cas, l'usage d'un logiciel par un utilisateur contribue à renforcer l'utilité que les autres utilisateurs vont retirer de cette utilisation, par un *effet de club* - également qualifié d'externalité positive par les économistes. Cet effet est le produit des interactions sociales entre les utilisateurs, par exemple lorsqu'ils désirent échanger des fichiers : il s'agit donc d'une valeur créée socialement, distincte de l'éventuelle valeur intrinsèque du logiciel.

Enfin, les logiciels ne peuvent pas fonctionner sans un support matériel, le "hardware". Dans l'histoire de l'informatique, "hardware" et "software" ont été initialement liés sur un seul marché, celui du hardware. Comme la simple possession du "hardware" ne permet pas, à elle seule, de produire des services utiles - les économistes disent des "caractéristiques d'usage"[3] - le software est un complément nécessaire pour la valorisation du hardware. En conséquence, la prolifération de logiciels permettant aux ordinateurs de produire des caractéristiques d'usage est de l'intérêt du vendeur de matériels. Dans ces conditions, la libre circulation du software entre les utilisateurs et son amélioration par ces derniers, seuls ou en partenariat avec les offreurs de matériel, a joué un certain rôle dans le développement de l'informatique naissante.

Production de logiciel et trajectoires technologiques diversifiées dans l'industrie informatique : des "mondes contrastés"

Pour les trois motifs évoqués ci-dessus, on peut dire que les questions soulevées par le mouvement des logiciels libres sont consubstantielles à l'informatique, et que les pratiques correspondantes sont nées avec cette industrie. Cependant, pour des raisons liées à des processus d'innovation technico-économiques tels que la division croissante du travail ou l'évolution des technologies disponibles, on a vu apparaître progressivement des logiciels conçus comme des produits indépendants destinés à être vendus sur le marché. Enfin, l'émergence de la mini-informatique et surtout de la micro-informatique s'est à son tour traduite par une mutation du logiciel en produit de masse : le "progiciel" (Mowery [1996]).

Cette autonomisation par étapes d'un marché indépendant du logiciel repose sur une base juridique, acquise par un statut d'éditeur, garanti par un copyright qui permet de privatiser un support trop aisément copiable. Par ailleurs, les techniques de compilation permettent de protéger

[2] Arrow suggère dans cet article que la rémunération de l'innovateur soit séparée des charges qui pèsent sur les consommateurs bénéficiaires de l'innovation. La recherche pourrait ainsi être financée par des organismes publics, mais également par des fondations soutenues par des acteurs privés. Ce type d'association sponsorisée tend à se développer dans le logiciel libre, parallèlement au bénévolat.

[3] On se réfère ici à une approche "Lancasterienne" de l'industrie informatique proposée par Delapierre M., Gerard-Varet L.A., Zimmermann J.B. [1980].

le code source (codification initiale des programmes). En pratique, les exécutables sont le plus souvent diffusés sans le code source par les vendeurs de logiciels. Parallèlement à ce processus de marchandisation du logiciel, la production coopérative avec la participation des utilisateurs de logiciel s'est toujours maintenue, et une démarche spécifique est même apparue autour de la distribution "Berkeley" du système d'exploitation Unix[4]. Cette démarche de coopération et de partage autour de l'Unix de Berkeley a joué un rôle important dans l'émergence du phénomène actuel des logiciels libres (McKusic [1999], Stallman [1999], Jullien, Phan [1999]).

La production du logiciel se déroule donc concurremment selon des logiques très différentes qui ont été analysées comme autant de "mondes" distincts, par exemple dans ce numéro, par Horn qui utilise une typologie générale en "mondes de production" proposée par Salais, Storper [1993] ou encore par Jullien qui interprète ces "mondes" comme autant d'arrangements techniques, économiques et socio-institutionnels spécifiques, reliés entre eux par les caractéristiques de leur trajectoire historique.

Dans cet article, nous insistons plus particulièrement sur les trajectoires technico-économiques qui ont conduit à l'émergence du mouvement des logiciels libres et sur les conséquences économiques des interactions et des apprentissages dans la production et l'innovation logicielle. Quelles caractéristiques économiques peut-on associer aux modalités de production - distribution de logiciel libre? Celles-ci forment-elles un système doté d'une cohérence interne, et comment celui-ci est-il articulable avec les autres modes de production - distribution du logiciel ? L'environnement a-t-il évolué d'une manière favorable à ces nouvelles pratiques ? En résumé, assistons-nous à l'émergence d'un nouveau modèle pour l'industrie informatique ?

Pour répondre aux différents aspects de cette question, on présentera dans une première partie un aperçu de l'historique de la production du logiciel vu sous l'angle de la participation des utilisateurs à cette production. La seconde partie rattachera les principes revendiqués par les producteurs de logiciels libres aux caractéristiques technico-économiques de la production et de l'innovation dans le domaine du logiciel, afin d'identifier les spécificités de ce mode de production de logiciels et d'en discuter la portée. La dernière partie évaluera les stratégies d'acteurs qui ont permis l'émergence du logiciel libre dans le monde marchand et proposera une évaluation de l'impact possible de cette évolution.

Le mouvement du logiciel libre comporte différents aspects, idéologique, politique, juridique et économique. C'est cette dernière approche qui nous intéressera principalement dans ce texte.

1. Eléments d'histoire de la production des logiciels non marchands

La participation des utilisateurs à la production de logiciels est un phénomène ancien et permanent, mais avec des formes historiques variables. Cette partie, est consacrée à la présentation de l'histoire de la production des logiciels à la lumière de ce phénomène. Elle se subdivise en quatre sous-parties qui correspondent chacune à autant de trajectoires historiques

[4] Pour les utilisateurs d'Unix (et de Linux) une "distribution" désigne un ensemble composé du noyau du système d'exploitation et d'une sélection d'utilitaires compatibles avec ce noyau. L'Université de Berkeley avait obtenu une licence d'AT&T qui lui permettait de modifier et de distribuer le système Unix. Les informaticiens de Berkeley ont ainsi ajouté des fonctionnalités au système initial, et les différentes versions de ces systèmes sont connues sous le nom de Distribution de Berkeley ou "BSD".

spécifiques. La première trajectoire correspond au domaine des mainframes et à la naissance de l'informatique. La seconde trajectoire peut être associée à l'émergence de la mini-informatique et d'une industrie du logiciel. Cette dernière ne prendra cependant sa forme actuelle qu'avec le décollage de la micro-informatique au début des années quatre-vingt. La troisième trajectoire caractérise l'environnement Unix et représente la voie hybride d'un système d'exploitation né du monde de la recherche et importé ensuite par les constructeurs dans le monde marchand. L'émergence dans les années 90 d'une trajectoire logiciel libre, autour de projets d'une version "libre" d'Unix et de logiciels associés à l'Internet peut être interprétée à la fois comme un prolongement des tendances apparues avec Unix et comme une réaction contre les formes de développement de l'industrie marchande du logiciel qui sont apparues dans le monde du PC.

1.1. Vingt-cinq ans de production de logiciel *" in house "*

L'informatique est une industrie née il y a 50 ans, et pourtant la question du logiciel n'a pas été posée pendant les 25 premières années, bien que celles-ci aient été caractérisées par une immense créativité logicielle. En fait, on peut considérer que la production de logiciel, en tant qu'activité marchande spécifique constituant un secteur distinct de la production de hardware, date du milieu des années 70, avec l'arrivée d'une nouvelle génération d'ordinateurs, le mini-ordinateur.

En effet, au début de l'informatique, la fabrication du logiciel était principalement le fait des utilisateurs eux-mêmes, aidés par les constructeurs et le monde académique. Le logiciel dit " de base " était fourni par les constructeurs et ne donnait pas lieu à facturation séparée. Le logiciel d'application était réalisé par les utilisateurs (entendons les informaticiens des organismes utilisateurs) avec une aide plus ou moins importante des fabricants de matériel. Dans le monde académique (universités et laboratoires de recherche), les programmes étaient assez librement échangés entre les différents producteurs / utilisateurs sans paiement de royalties. Bien qu'elle soit marginale, une tradition de diffusion non marchande de logiciel s'instaure donc dès le début de l'informatique.

Le continuel élargissement des tâches effectuées avec l'outil informatique, permis par la baisse des prix[5], a entraîné un besoin de plus en plus important de logiciels, qu'utilisateurs et constructeurs ne pouvaient continuer à produire seuls. C'est alors que sont nées les sociétés de service et de conseil en informatique (SSCI) et quelques éditeurs de logiciels spécialisés dans les outils pour grands et moyens systèmes, tel *Computer Associates*. Mais la segmentation du marché imposée par les constructeurs et leurs systèmes d'exploitation propriétaires ne poussait pas les SSCI, de plus en plus actives dans la fabrication de logiciel, à transformer les développements qu'elles réalisaient pour leurs clients en *produits logiciels* puisqu'il fallait, de toute façon, les adapter aux matériels différents et aux spécificités des clients qui étaient habitués au "sur mesure".

Le logiciel n'était pas encore un produit marchand, dans la mesure où il n'était pas destiné à être répliqué pour être vendu sur un marché, et qu'il était le plus souvent produit par les utilisateurs eux-mêmes (seul, ou en coopération avec des constructeurs ou des sociétés de services) pour répondre à des besoins internes à leur organisation.

[5] Les gains de productivité sont de 20% par an sur 40 ans, Genthon, [1995].

1.2. L'industrie marchande du logiciel et la création des progiciels

La naissance d'une industrie de logiciel est liée au développement de la mini-informatique. D'une part, ce nouvel outil informatique permettait d'atteindre des entreprises qui avaient de faibles compétences informatiques et étaient donc prêtes à acheter des programmes d'application. D'autre part, la mini-informatique a permis le développement d'activités plus spécialisées comme la CAO, pour lesquelles des solutions logicielles de type programmes achetés à des fournisseurs de logiciels étaient bien adaptées[6].

L'apparition de la micro-informatique au milieu des années 70 a accentué le phénomène de développement d'une industrie indépendante du logiciel. Ces produits avaient besoin de systèmes d'exploitation, d'outils et d'applications. D'où la naissance des fabricants de systèmes d'exploitation pour micro (par exemple *Digital Research* et *CP/M*), d'outils (par exemple *Microsoft* et son *Basic*) et surtout l'innovation fondamentale de la "feuille de calcul" - ou "tableur" - par Dan Bricklin (*VisiCalc*). Il s'agit une nouvelle forme d'application dotée de caractéristiques génériques qui permettent à l'utilisateur non informaticien de mettre lui-même en œuvre la solution qui correspond à son besoin. S'inventent alors les *progiciels*, produits logiciels dont les caractéristiques d'usages sont assez générales pour leur permettre d'être utilisés par un grand nombre de personnes. La micro-informatique est aussi l'occasion de la diffusion de " petits " logiciels gratuits ou vendus au coût marginal, les *freeware* ou les *shareware*, par l'intermédiaire des revues spécialisées. Précisons que ces logiciels sont en général diffusés en version compilée et donc ne sont donc pas libres au sens défini dans cet article.

Jusqu'à l'apparition de la micro informatique, l'industrie informatique était caractérisée par une intégration matériel/logiciel de base qui étaient la propriété du constructeur de matériel. On parle alors de systèmes propriétaires. Le développement de la micro-informatique a rapidement vu la domination d'un système d'exploitation, MSDOS de Microsoft, qui équipait dès le milieu des années 80 plus de 80% des PC, introduisant une standardisation de fait. MSDOS était la propriété de Microsoft qui a alors dicté l'évolution technico-économique du monde de la micro-informatique. Voilà pourquoi on peut parler de standardisation propriétaire (Genthon, [1995], [1998]). Cette standardisation propriétaire autour de la micro-informatique (le PC associé à *Microsoft* et le Mac d'*Apple*) et les baisses de prix qu'elle induit, a permis dans les années 80 de mettre en place une nouvelle informatique fondée sur l'utilisation individuelle et systématisée de progiciels.

1.3. Unix, un positionnement intermédiaire

Le système d'exploitation Unix a été développé à partir de 1974, initialement sur le mini-ordinateur PDP11 de Digital, par des chercheurs des Bell Labs, K. Thompson et D. Ritchie. L'objectif était de réaliser un environnement système multitâches et multi-utilisateurs proche de *Multics*[7], mais plus simple, et adapté aux mini-ordinateurs. Parallèlement, D. Ritchie et B. Kernighan ont conçu un langage de haut niveau, le langage "C", qui rend Unix assez facilement portable sur différents matériels. A la suite du "*Consent Decree*" de 1956, AT&T était soumis à une stricte limitation de ses activités au domaine réglementé du téléphone, et s'interdisait donc toute activité marchande dans des domaines tels que les composants ou l'informatique. En

[6] Pour une présentation du développement de l'industrie du logiciel autour de la mini-informatique dans la région de Boston, voir Haas, [1995].

[7] Le système d'exploitation le plus avancé de l'époque, conçu par General Electric et le MIT avec la participation des Bell Labs.

conséquence, la licence d'Unix fut proposée à des conditions très libérales à l'Université de Berkeley : celle-ci dispose en effet du code source et de la possibilité de le modifier et de le redistribuer, le système originel restant cependant propriété d'AT&T. Berkeley va rapidement développer une distribution d'Unix appelée BSD qui s'est largement diffusée dans le monde académique[8]. Unix sortira lentement de sa niche universitaire aux débuts des années 80 grâce aux stations de travail et aux applications scientifiques. C'est en effet le système d'exploitation choisi par Sun pour se positionner sur ce marché. Son succès entraîne ses concurrents à l'imiter. Mais chaque firme propose sa propre implémentation d'Unix. Il existe donc un problème de compatibilité au sein même de la famille des systèmes Unix, et une demande d'harmonisation de la part des utilisateurs. Développé par Sun, NFS (*Network File System*), un logiciel qui permet des échanges de fichiers entre stations de travail de constructeurs différents connaît ainsi un large succès.

L'histoire de la famille des systèmes Unix est aussi celle de l'échec des tentatives d'unification. Le démantèlement d'AT&T en 1984 permet à la firme de se lancer dans l'industrie informatique, et cela se traduit par une tentative de reprise en main d'Unix. AT&T devenant un acteur informatique, l'accord conclu en 1987 avec Sun pour faire converger Unix System V (AT&T) et Sun OS (Unix Berkeley 4.3) a été interprété comme une volonté de se (ré)approprier le développement futur d'Unix. La réponse des concurrents a pris la forme de la création de l'Open Software Foundation en 1988 par un association de constructeurs (Appollo, Bull, Digital, HP, IBM, Nixdorf, Philips et Siemens) pour produire des logiciels intégrables dans les plate - formes des membres du groupes. La volonté d'assurer l'interopérabilité des applications dans l'univers Unix et de favoriser leur développement s'est traduite par la création du groupe X/Open par les constructeurs européens bientôt rejoints par les constructeurs américains et par de grands utilisateurs. Ce groupe rédigea un certain nombre de recommandations pour assurer la portabilité des applications.

Ces initiatives ont ainsi introduit la voie des "systèmes ouverts", par des interfaces standardisées[9], comme remède à la diversité des solutions propriétaires. Mais elles eurent des résultats limités. Les environnement de la famille Unix sont largement restés une collection de systèmes propriétaires proposés par les offreurs de grands systèmes, de mini-ordinateurs et de stations de travail. Chaque constructeur a maintenu sa spécificité pour se différencier. Ce système était viable tant qu'Unix n'avait pas de concurrents dans son domaine. Mais ce n'est plus le cas aujourd'hui et Microsoft, profitant de la balkanisation des environnements Unix, propose depuis 1992 un système d'exploitation concurrent, Windows NT. Celui-ci est bien sûr propriétaire comme tous les produits de Microsoft. Mais il a pu s'imposer grâce à sa proximité avec la famille Windows 9X et à la puissance de Microsoft. Comme nous le verrons dans la suite, c'est cette invasion par Microsoft du terrain des moyens et grands systèmes qui pourrait offrir au logiciel libre une nouvelle opportunité de développement.

Mais l'ère des Unix propriétaires a également fourni une incitation pour l'émergence d'un

[8] BSD était cédée à un prix voisin de 1000 $, avec la possibilité de réaliser des copies des exécutables et de modifier le code source (McKusic [1999]).

[9] Le terme "systèmes ouverts" désigne ici des systèmes qui sont "ouverts" par certaines interfaces dont les spécifications sont publiques pour permettre la compatibilité avec d'autres logiciels rendant des services complémentaires. La spécification publique d'une interface ou d'une architecture peut, à son tour, être "propriétaire" (comme par exemple DCOM = Distributed Composant Model de Microsoft) ou normalisée par consensus dans un comité *ad hoc* (comme par exemple CORBA = Common Object Request Brocker, normalisé par l'Object Management Group).

mouvement de "libération" du logiciel, personnifié par Richard M. Stallman, qui a conçu initialement son projet GNU pour promouvoir un système d'exploitation, clone d'Unix, libre.

1.4. Les logiciels libres, Linux et Internet

En 1985, R. Stallman crée la *Free Software Foundation* (FSF) pour recueillir des fonds en faveur du développement de son projet alternatif. Il définit et consolide juridiquement la notion de logiciel libre. Un logiciel est dit *libre* si l'utilisateur peut exécuter, modifier librement le programme et le distribuer à nouveau dans les mêmes conditions sans pouvoir ajouter des restrictions supplémentaires à ce mode de diffusion (copyleft). Cette dernière condition vise à protéger un logiciel libre contre une privatisation toujours réalisable à partir d'un logiciel du domaine public. Stallman [1999] cite l'exemple de *X-Windows*, initialement distribué par le MIT avec une licence très permissive, qui a été approprié par des promoteurs des environnements Unix propriétaires, qui l'ont intégré à leur système sous forme de codes binaires exécutables, diffusés sans le code source. En pratique, un logiciel libre doit donc être fourni avec son code source. La FSF a développé une licence permettant de protéger le logiciel libre, la General Public Licence (GPL).

Mais le projet GNU s'est développé lentement par manque de ressources. Parallèlement, d'autres projets d'Unix libres pour architecture PC ont été mis en oeuvre, et finalement, c'est le système Linux, lancé par Linus Torvalds qui s'est le plus largement diffusé, sans doute initialement pour des raisons en partie contingentes[10], mais surtout grâce à une méthode nouvelle de développement de logiciel utilisant largement les possibilités de travail coopératif offertes par l'Internet.

Les conditions GNU étant assez restrictives, la diffusion des logiciels libres dans le système marchand peut être facilitée par le relâchement d'un certain nombre de contraintes [11]. C'est la voie choisie par le mouvement *Open Sources*, qui demande à ses membres de fournir le code source de leur logiciel tout en permettant de conserver le copyright [12]. Les utilisateurs peuvent faire des modifications privées mais non publiques des logiciels. Le concept Open Sources permet ainsi de faciliter les relations avec le logiciel propriétaire. Aujourd'hui, un certain nombre d'éditeurs traditionnels de logiciels ont adhéré à Open Sources, et d'autres on adopté d'autres formes licence pour ouvrir leur code source (Netscape, par exemple), et il ne fait aucun doute que le regroupement autour de cette approche va dorénavant être une référence significative pour les producteurs de logiciels.

L'outil qui a favorisé le développement technique des logiciels dont le code source est ouvert est **Internet**. Au début de son histoire, Internet était un réseau d'informaticiens et de chercheurs

---

[10] La thèse de la réussite contingente est soutenue par R. Young, fondateur de Red Hat Sofware, le principal diffuseur commercial de Linux.

[11] Nécessité déjà perçue par Stallman, qui distingue plusieurs types de licences : la GNU Library GPL autorise l'utilisation de cette bibliothèque par des logiciels propriétaires. Il s'agit d'un choix stratégique, selon Stallman [1999] : "limiter l'utilisation de notre bibliothèque du langage C au logiciel libre n'aurait donné aucun avantage au logiciel libre; cela n'aurait eu pour effet que de décourager l'utilisation de notre bibliothèque. [...] Il n'existe aucune raison éthique d'autoriser des applications propriétaires sur le système GNU, mais d'un point de vue stratégique, il semble que les interdire découragerait plus l'utilisation d'un système GNU que cela n'encouragerait le développement d'applications libres".

[12] Cf. la définition de "Open source" (www.opensource.org) et sa traduction française avec une justification : (www.linux-france.org/article/these/osd/fr-osd.html) pour une critique de ce modèle d'ouverture par Stallman : Pourquoi ``Free Software" est-il meilleur que ``Open Source" ? : www.april.org/groupes/gnufr/open.html

financé par le gouvernement fédéral américain (ARPA, NSF). Les principaux outils de l'Internet (les serveurs les plus utilisés, comme Apache, Sendmail etc., une partie des logiciels clients, navigateurs, messageries etc.) appartiennent eux-mêmes majoritairement au monde du logiciel libre. Pour notre propos, Internet permet des échanges de fichiers informatiques à travers le monde d'une manière simple et facilite le travail coopératif à distance. C'est cette fonctionnalité qui va être saisie par les informaticiens pour développer une méthode originale de fabrication du logiciel. Internet a fait franchir à la production de logiciel libre un seuil critique. En effet, les interactions entre producteurs et utilisateurs qui permettent le partage des connaissances existaient déjà dans le monde Unix mais elles étaient forcément très localisées. Le développement de l'Internet permet une multiplication et une globalisation de ces interactions.

2. Economie du développement de logiciels et modèle économique du logiciel libre

Quels sont les fondements économiques du développement des logiciels libres ? On présentera en premier lieu les conditions de fabrication des logiciels (§ 2.1). On étudiera ensuite les conditions de production spécifiques des logiciels libres. Deux axes sont plus particulièrement développés dans cette partie. Au mode de développement de projets à coordination centralisée, on oppose un mode de développement collectif plus décentralisé dont on essaye de qualifier les propriétés économiques (§ 2.2). On souligne ensuite le rôle des interactions utilisateurs - développeurs et des externalités associées à la circulation du code source (§ 2.3).

2.1. Production, édition et distribution du logiciel

Cette sous-partie compare la production des mondes marchand et libre à partir d'une analyse de l'économie du progiciel.

Une représentation simple de l'économie des progiciels peut être envisagée autour de trois activités : la production, l'édition et la distribution. La production du logiciel correspond aux activités de spécifications, conception, codage, intégration et maintenance. L'édition correspond aux activités de packaging (duplication sur supports divers, édition des manuels, etc.) et la distribution aux activités de mise à disposition du produit aux utilisateurs. Une quatrième activité est le service après vente mais, dans le monde des progiciels, il est assez souvent sous-traité.

Quelles sont les caractéristiques économiques de cette activité? La production correspond à un coût fixe car le logiciel doit être produit avant que d'être vendu et quel que soit le nombre futur de copies vendues. Les coûts d'édition sont faibles car les coûts de duplication se réduisent aux coûts du ou des supports (bande, disquette, CDROM, livre). La distribution est assurée par les circuits de distribution généralistes et spécialisés, parfois complétés par la vente directe, (notamment par correspondance). Les coûts associés sont traditionnels (une proportion du prix de vente). Notons les barrières à l'entrée imposées par des réseaux de distribution assez concentrés qui disposent d'un important pouvoir de négociation dans les rapports éditeurs - distributeurs.

Globalement, l'économie du progiciel est caractérisée par des coûts fixes élevés par rapport au coût total. La théorie économique nous indique que, dans ce cas, une structure oligopolistique ou une concurrence monopolistique peut émerger. Le champ des applications informatiques étant

très large, la concentration existe par domaine mais pas globalement.

Comment peut s'insérer dans ce contexte le logiciel libre ? Prenons les trois domaines décrits précédemment:

Le financement du coût fixe de production peut être assuré par des producteurs appartenant à la sphère publique (Université, Recherche Publique), par le travail non payé des " hackers ", par les voies du sponsoring (comme la FSF de Stallman, par exemple) ou encore par des éditeurs ou distributeurs (comme Red Hat).

L'édition des logiciels libres est aujourd'hui réalisée par des sociétés de service qui assurent souvent la fonction de distribution, quand celle-ci est organisée (un certain nombre de logiciels libres sont disponibles directement sur Internet).

La distribution et des services associés sont un élément clef pour la diffusion et le financement des logiciels libres. La complémentarité entre les services associés au logiciel (prestations de conseil et d'assistance) et le logiciel libre lui-même est, en effet, un des arguments utilisés par les partisans du "libre" (Smets-Sloanes, Faucon [1999], Behlendrof [1999]). Selon ces auteurs, comme les clients n'ont plus à supporter le coût d'acquisition, ils peuvent consacrer plus de ressources aux dépenses en services. Enfin, la gratuité du logiciel devrait favoriser une plus grande diffusion.

L'originalité des logiciels libres se manifeste surtout dans une plus grande décentralisation des activités de production et dans les interactions utilisateurs - développeurs.

2.2. Coordination centralisée versus décentralisée

La production classique de logiciel a été formalisée dans un modèle de développement (le *waterfall*) (Schach, [1990]). L'organisation est centralisée et hiérarchisée et les différentes étapes sont temporellement ordonnées. Notons que Microsoft a " parallélisé " cette méthode et que, de plus, les chefs de projets ne fixent les spécifications qu'au tout dernier moment. Cela permet un gain de temps appréciable mais a la conséquence bien connue d'augmenter les *bogues* des logiciels de la firme.

Linux a popularisé une nouvelle méthode de fabrication dont les principes sont décrits par Raymond dans une contribution intitulée "la cathédrale et le bazar" (Raymond [1998]). On la qualifiera de *développement distribué*. Le principe peut se résumer ainsi: un projet doté d'un animateur est lancé sur Internet et de très nombreux programmeurs participent de manière bénévole et informelle aux activités de fabrication (le bazar).

Selon la " loi de Brooks "[13], la gestion de projet centralisée (la cathédrale) est caractérisée par des rendements décroissants à l'échelle, attribuables à la complexité du projet. Au-delà d'un certain seuil de complexité, une méthode décentralisée, fondée sur la sélection compétitive de solutions locales mises en concurrence, pourrait ainsi devenir plus efficace qu'une solution centralisée. Des exemples qui illustrent ce phénomène peuvent être observés en intelligence artificielle, où des méthodes décentralisées de résolution de problème par systèmes "multi-agents" peuvent se révéler d'autant plus efficace relativement aux méthode centralisée que la complexité augmente (Cf. Bersini, Kuntz, Syers, [1994]).

La programmation du code, étape préférée des programmeurs, permet par ailleurs de tirer

[13] "les coûts de communication et de complexité d'un projet (supervisé) augmentent de manière quadratique avec le nombre de développeurs, alors que le travail réalisé n'augmente que linéairement " (cité par Raymond [1998]).

parti de leur créativité et de la décentralisation du processus. De même, la contribution décentralisée des programmeurs et de certains utilisateurs au débogage apparaît comme le second point fort du processus. C'est alors la *nature des interactions* qui devient centrale, comme le souligne Raymond ([1998] §10 : le contexte social).

Dans les autres domaines, le réseau Internet permet d'atteindre une masse critique pour la diffusion et pour la recherche de main d'oeuvre qualifiée prête à contribuer bénévolement (que l'on devrait trouver plus généralement dans le monde public ou dans les grandes organisations, mais aussi peut être de manière transitoire chez les prestataires de services).

Les points faibles de la méthode concernent les spécifications et la documentation. En effet, c'est la première étape (les spécifications) qui est *a priori* la plus difficile à initier de manière décentralisée, comme le soulignent par exemple Raymond [1998] ou Behlendrof [1999]. Quant à la documentation, c'est traditionnellement un point faible de tous les programmeurs, qu'ils soient hackers ou non. Mais le adeptes du libre ne disposent pas des incitations qui existent dans le monde marchand. La faiblesse au niveau des spécifications peut cantonner le mode de production décentralisé à des domaines assez stabilisés, ou bien nécessiter l'introduction d'un certain niveau de coordination, en particulier au démarrage du projet, comme le suggère Raymond [1998]. En ce qui concerne la documentation, cette partie peut être prise en compte au niveau des moments édition / distribution.

La décentralisation peut être également interprétée comme la possibilité - ou plus exactement la liberté - de mettre en œuvre une pluralité de solutions en concurrence sur le mode de la "R& D concurrentielle ". Dans ce cas, la solution la plus adaptée va s'imposer et le modèle compétitif du "bazar" peut être considéré comme supérieur au modèle marchand centralisé qui est dominé, nous l'avons déjà noté, par une structure peu concurrentielle. La liberté est alors également celle d'entreprendre[14]. La nature modulaire de l'architecture logicielle contemporaine limite cette concurrence "intra-libre" à des sous-modules biens identifiés supportant généralement les mêmes fonctionnalités[15].

2.3. Interactions utilisateurs - développeurs

Rosenberg ([1982]) a depuis longtemps souligné l'intérêt de ces interactions pour la mise en oeuvre d'innovations adéquates aux besoins des utilisateurs. Dans les logiciels libres, de telles interactions peuvent exister lors de l'étape de définition des besoins, en particulier grâce au groupes de discussions, comme le souligne Vixie [1999][16], mais surtout pour la phase de débogage.

"traiter vos utilisateurs en tant que co-développeurs est le chemin le moins semé d'embûches

---

[14] Ce discours libertarien, aux accents à la fois libertaires et libéraux, se retrouve chez certains représentants du logiciel libre, qui éprouvent parfois le besoin de chercher des fondements économiques à leur vision du monde, comme c'est le cas par exemple pour Raymond avec Hayeck. Pour une discussion de ces question, Cf. Jullien, Phan [1999].

[15] Cette concurrence entre sous-modules supportant des fonctionnalités comparables n'exclut d'ailleurs pas la recherche d'un certain degré de complémentarité : par exemple, l'interface graphique Gnome permet de "récupèrer" un certain nombre de modules de son concurrent "KDE".

[16] "Les gens du logiciel libre ont tendance à développer les outils dont ils ont besoin ou dont ils ont envie. Il arrive quelquefois que cela coïncide avec ce qu'ils font au travail, et souvent ce sont ceux dont la tâche première est l'administration plutôt que le développement de logiciel. [...] La discussion d'un Document de Spécification d'un Logiciel Libre ("DSLL") se situe généralement au sein d'une liste de diffusion ou d'un groupe de discussion où les utilisateurs et développeurs discutent directement".

vers une amélioration rapide du code et un débogage efficace" (Raymond, [1998]).

La diffusion du code source permet à tout utilisateur - développeur d'intervenir sur le logiciel et de produire des adaptations, améliorations, compléments, etc. Ces modifications, qui sont facilement diffusables sur Internet pourront elles-mêmes être adaptées ultérieurement par d'autres. Un processus d'apprentissage collectif se met ainsi en place, doublé d'un mécanisme de sélection compétitive des solutions fondé sur leur intérêt pour les utilisateurs - développeurs.

Ce système est particulièrement adapté à une communauté d'utilisateurs - programmeurs. Peut-il fonctionner à grande échelle avec le passage à une large diffusion ? Le développeur coopératif a été caractérisé comme plus disposé à s'investir intellectuellement que financièrement. Mais dès lors que l'on peut disposer librement d'améliorations produites par le travail d'une minorité, pourquoi la majorité s'investirait-elle ? Il s'agit d'un problème économique classique de " passager clandestin ". Le modèle d'incitation le plus fréquemment cité est le modèle universitaire de la reconnaissance par les pairs. Hors de la sphère académique, cette incitation, même si elle peut exister, doit être mise en balance avec l'affectation du temps consacré à cette activité au détriment des autres activités de l'organisation utilisatrice. De plus, dès lors que le " libre " sort du monde des utilisateurs - développeurs où il était jusqu'alors confiné, une proportion croissante d'utilisateurs se comportant en clients passifs - et avec de nouvelles exigences - vont apparaître. Les acteurs de la distribution pourront-ils assumer seuls cette responsabilité ?

Globalement, il semble que le modèle du libre ne subisse ni désavantages ni, par contre, ne bénéficie d'avantages par rapport au modèle marchand. En effet, si le moment de la production profite du travail coopératif gratuit et des synergies utilisateurs - développeurs, les moments de l'édition et de la distribution sont plutôt désavantagés (problèmes de commercialisation et de bouclage offre / demande, etc.). En fait, si le libre se développe aujourd'hui, c'est aussi parce qu'il s'insère dans les stratégies des grands acteurs du monde informatique.

3. Stratégies d'acteurs

L'économie du logiciel libre est largement fondée sur le modèle suivant : je donne le produit qui est développé en dehors du système marchand et je vends du service autour de ce produit. Cette économie, qui caractérise plus particulièrement les distributions Linux, est-elle généralisable à beaucoup de logiciels ? Est-ce uniquement du "service" qui est vendu ou bien du service et une partie de production de logiciel qui n'est pas dans le domaine public ? Par exemple, Red Hat assure une activité de plus en plus conséquente de développement. Celle-ci se limitait à l'intégration des différents modules hier. Elle participe au développement d'une interface conviviale aujourd'hui.

Les logiciels dont le code source est ouvert, existent depuis le début de l'informatique mais restaient cantonnés aux marges du système marchand. Cette fin de siècle voit le logiciel libre devenir une alternative plus crédible aux systèmes propriétaires. L'évolution actuelle est si rapide qu'il faut chercher des explications complémentaires en dehors du libre lui-même, bien que le couple Linux / Internet ait joué un rôle central dans l'émergence ce phénomène, comme nous l'avons souligné. La réussite possible des logiciels libres tient à la configuration de l'industrie

informatique (matériel et logiciel) qui est en voie d'être dominé par Microsoft. Le regroupement autour du libre devient la stratégie des "David" contre le "Goliath ".

3.1. La domination de Microsoft : une incitation à rallier le monde du libre ?

Microsoft est un monopole en expansion qui utilise tous les moyens possibles pour dominer le monde du logiciel (Di Cosmo, Nora, [1998]). A partir de sa base de PC, il remonte sur le marché des serveurs avec Windows NT, profitant de la fin des systèmes propriétaires et de la faiblesse du monde Unix "classique" qui ne peut plus présenter des avantages en terme de puissance par rapport au machines construites autour de processeurs Intel[17] et qui est divisé. Jouant sur les avantages que lui procure la maîtrise du système d'exploitation, Microsoft est devenu également dominant en bureautique et essaie de s'imposer sur Internet. Les méthodes de Microsoft, rachat à faible prix des sociétés qui réalisent des innovations ou copie pure et simple de leur technologie rendent plus difficiles la naissance, le développement ou quelquefois la survie des sociétés de logiciel indépendantes. En termes économiques, Microsoft a fortement élevé les barrières à l'entrée dans l'industrie. D'une part, ses méthodes prédatrices freinent l'engagement du capital risque à s'investir dans le secteur, d'autre part, en augmentant sans cesse les fonctionnalités des logiciels grand public et professionnels il ne laisse que peu de niches à ses concurrents pour survivre. Avec Windows, Microsoft restait limité au monde des PC, monde qui représente la majorité mais pas la totalité du marché informatique. Avec Windows NT, Microsoft s'attaque au coeur du marché de tous les grands constructeurs informatiques.

Le logiciel libre abaisse les barrières à l'entrée et c'est une raison importante de son succès. Il prend son essor au moment où l'ensemble des acteurs de la production de logiciel mais aussi de la production de matériel, craignant que Micrososft ne les étouffe à terme, y voient l'occasion de combattre sa position hégémonique et retrouver des marchés qu'ils avaient perdus. En effet, il est de l'intérêt de l'ensemble des acteurs (constructeurs de matériels, fabricants de logiciels, utilisateurs) qu'une architecture alternative se développe. C'est précisément ce que propose le monde libre autour de Linux. C'est pour cela que ce monde a obtenu le soutien de Netscape, en opposition frontale avec Microsoft pour les logiciels de navigation, de Corel en opposition sur les suites bureautiques. et de constructeurs comme IBM qui subissent aujourd'hui les conditions léonines de Microsoft. C'est pour cela encore qu'Intel soutient Red Hat - et donc Linux - car il a, lui aussi, tout intérêt à ce qu'une alternative logicielle à Microsoft se développe autour de ses processeurs.

Aujourd'hui, la position dominante de Microsoft, qui pèse sur tous les acteurs, incite ces derniers à soutenir une solution alternative. Et c'est précisément parce que cette solution vient d'un autre monde qu'elle peut susciter le consensus d'acteurs en concurrence (rappelons les échecs d'unification d'Unix).

3.2. Premiers effets du développement des logiciels libres

Dans le domaine des systèmes d'exploitation, il semble acquis que Linux dispose d'un véritable marché, mais celui-ci est pour l'instant encore largement cantonné aux marchés des systèmes pour serveurs Internet. C'est du moins le seul segment où Linux obtient une part de

[17] Il se vend 100 fois plus de PC que de stations de travail et les économies d'échelle en production de composants font que les processeurs d'Intel rattrapent inexorablement les processeurs Risc des stations de travail. Il faudrait une nouvelle révolution architecturale, comme le Risc par rapport au Cisc, pour que les stations de travail puissent se démarquer ... mais on ne la voit pas venir.

marché significative (Cf. tableau). Il bénéficie ainsi, au même titre que Windows NT de la montée en puissance des PC face aux stations de travail bas de gamme.

Les premières victimes de l'expansion de Linux sembleraient être plutôt du côté du monde Unix que de chez Microsoft (selon la logique du maillon le plus faible). Ceci apparaît lorsque l'on regarde les parts de marchés des différents systèmes d'exploitation supportant les serveurs Internet. Il reste à savoir si cette logique s'appliquera, à terme, à un nombre significatif d'autres segments du marché.

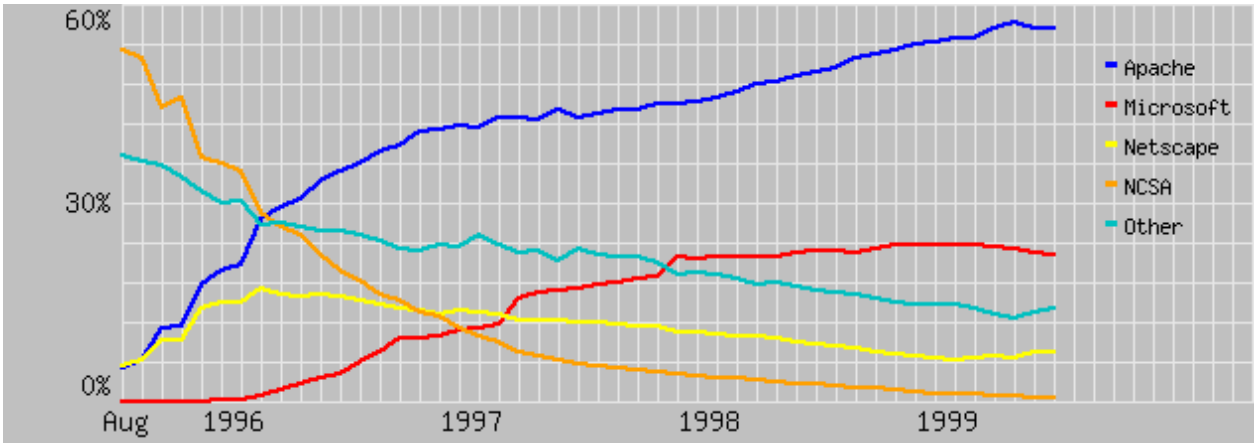
Tableau : parts de marché des Systèmes supportant les serveurs Internet en avril 1999

	Services : http + ftp + news		http ; tous domaines	
	Part de marché (avril 1999)	Evolution de janvier à avril en part de marché	Part de marché (avril 1999)	Evolution de janvier à avril en part de marché
Linux	24.3%	+ 2.8%	28.7%	+ 2.0 %
Windows 9X/NT	18.9%	- 0.1%	24.6%	+ 0.3 %
SunOs Solaris	16.7%	- 1.%	18.9%	-1.2 %
Unix BSD family	14.6%	- 0.4%	15.3%	- 0.3 %
IRIX	4.6%	- 0.7%	3.8%	- 0.3 %
Mac/Apple	2.1%	+ 0.5%	2.2%	+ 0.5%
AIX	1.4%	- 0.3 %	1.4%	- 0.2 %
Autres Unix *	2.5%	- 0.8 %	2.6%	- 0.9 %

Source : The Internet Operating System Counter : www.leb.net/hzo/ioscount/index.html
*Autres Unix : HPUX; Reliant Unix/Sinix; Digital Unix; SCO Unix

Sur le marché des serveurs Web, il est clair que l'offre conjuguée d'un logiciel libre et gratuit, Apache (multi-plates-formes) et d'un logiciel gratuit mais non libre (IIS de Microsoft, qui tourne dans l'environnement WindowsNT) a définitivement ruiné le marché des petits éditeurs qui cherchaient à vendre leurs serveurs (l'exemple le plus notable restant Nescape) comme le montre la figure ci-dessous. Ces derniers sont pris en ciseaux par la stratégie de gratuité de Microsoft à la fois sur le marché des serveurs et sur le marché des navigateurs et suites Internet. Après avoir essayé quelques mois de faire payer son logiciel client, Netscape a du y renoncer. Il le distribue à nouveau gratuitement, choisissant du coup d'ouvrir le code source de son logiciel et de trouver des financements indirects de son activité (rachat par AOL, partenariat avec Sun).

Figure : Part de marché des logiciels serveurs http



source : http://www.netcraft.co.uk/Survey

Par contre, Microsoft devrait encore bénéficier de larges marges de manoeuvre dans le futur, même s'il est contraint de baisser significativement ses prix et donc ses marges. Premièrement, les capacités d'absorption technologiques de la firme de Redmont sont telles que les innovations qui pourraient apparaître dans le domaine du libre contribueront certainement à l'amélioration de la qualité et des fonctionnalités des prochaines versions de son offre. Deuxièmement, l'émergence d'une concurrence au niveau des logiciels systèmes sera exploitée par la firme dans la procédure antitrust auquel elle est exposée. Enfin, pour sortir des niches de marché ou il est le mieux implanté (serveurs, enseignement, recherche, administrations), il reste encore beaucoup à faire à Linux dans le domaine de la convivialité et de la facilité d'installation des composants matériels, en particulier en ce qui concerne le grand public[18]. Microsoft pourrait ici encore tirer son épingle du jeu, y compris en proposant une version de ses applications bureautiques pour Linux!.

L'émergence de Linux représente une opportunité pour certaines catégories producteurs de logiciels non libres fonctionnant également dans l'environnement Windows, qui ont choisi de se rendre compatible avec Linux. Ainsi en est-il de nouveaux entrants dans le domaine des suites bureautiques avec *StarOffice* de Star Division, racheté récemment par Sun (qui annonce la possibilité d'accéder au code source… mais à quelles conditions ?), ou de firmes plus anciennes, comme Corel, fragilisé par la concurrence des produits Microsoft, qui donne une nouvelle jeunesse à *Wordperfect* en le rendant disponible sous Linux.

Selon la même logique, le libre pourrait également s'imposer dans certains sous-secteurs les moins concentrés, comme les utilitaires. En effet, la disposition sur Internet d'un ensemble d'outils de développement (éditeurs, compilateurs, etc.) rend la position des éditeurs qui vivaient auparavant de ces marchés, très difficile. En fait, les *petits éditeurs* sont déjà les premiers touchés par le monde du libre et la mouvance Internet.

Deux autres impacts du logiciel libre doivent être notés. D'une part, le débat sur les logiciels libres tend à évacuer un autre débat qui est celui des *standards ouverts* promus par les instances internationales de normalisation. Les logiciels dont le code source est ouvert pourraient

[18]La sortie au printemps 1999 de la version 6.0 de Redhat, avec les deux interfaces graphiques KDE et Gnome montrent que les choses bougent du coté de Linux. Une course est désormais engagée entre les deux systèmes.

renforcer, dans le futur, les standards ouverts (par exemple CORBA contre DCOM de Microsoft) si l'on en croit les orientations récentes des développeurs de Gnome et de KDE qui évoluent vers CORBA[19]. Le logiciel libre pourrait ainsi aider à renforcer les standards ouverts qui permettent à la concurrence de jouer plus efficacement.

Le second impact du logiciel libre concerne les conditions institutionnelles de protection des droits de propriété dans le domaine du logiciel. En effet, si l'ouverture du code source semble devoir s'élargir, on assiste également au développement des brevets, compatibles avec un code source public mais en fait plus restrictifs. Le mouvement se développe aux Etats-Unis et les brevets sont assez largement utilisés par Sun sur des séquences de code. Mais même dans ce cas, il existe des types de brevets rendant possibles une réutilisation, par exemple pour produire des variantes (Foray [1995]). Ici, c'est le monde du logiciel libre qui a ouvert le débat copyrigt / brevets.

CONCLUSION

Le logiciel libre se trouve aujourd'hui partie prenante du mouvement d'évolution de l'industrie du logiciel qui est encore dans sa phase de démarrage. Produits, modes de production et modes de distribution ne sont pas encore totalement stabilisés. Entre les logiciels libres sous licence GPL et les logiciels marchands traditionnels, de nombreuses formes existent comme par exemple les logiciels Open Sources dont nous avons parlé mais aussi toutes les formules de distribution inventées autour d'Internet (par exemple donner la version n-1 d'un logiciel et faire payer la version n, etc.).

Dans ces conditions, on peut imaginer un triple impact du logiciel libre sur une industrie du logiciel en devenir.

- généraliser l'ouverture du code des logiciels (diffusion du code source), ce qui peut être d'importance pour les informaticiens, mais d'un intérêt direct faible pour les utilisateurs non informaticiens.

- offrir une alternative crédible à Microsoft et pour cela accélérer la disparition des architectures et des systèmes d'exploitation propriétaires et des Unix non libres, hors Microsoft.

- instaurer avec et autour d'Internet un espace de réflexion et de discusion sur le devenir du logiciel.

Une seconde remarque concerne le modèle économique du libre. La production n'a réussi à dépasser le cercle scientifique que grâce à la méthode Linux. La généralisation de ce modèle dépend largement du bon vouloir de "sponsors" - publics ou privés - qui se substitueront au capital risque ou au monopole pour financer les coûts fixes de développement nécessaires pour lancer de nouveaux logiciels et pour accélérer leur développement (par exemple, contribution de programmeurs financés par Red Hat au développement de l'interface graphique Gnome). Quant à la distribution, elle reste largement standard, même si le recours à Internet est possible. Selon la logique du "service" les distributeurs devraient également affronter le problème de la maintenance, en particulier en ce qui concerne les entreprises. La politique de prix, elle, reprend

[19] Linux France Magazine, N° 8 juillet/aout 1999 pp 16-17 sur CORBA et DCOM.

les ingrédients classiques, mais à un niveau sensiblement plus faible (les distributions Linux sur CD les plus populaires coûtent de 4 à 10 fois moins cher que Windows 98 - et donc beaucoup moins cher que Windows NT). En fin de compte, une conséquence appréciable du développement des logiciels libres aura déjà été de permettre à de nouveaux acteurs d'entrer sur le marché et donc de faire baisser les prix. Ce qui est déjà bien !

Une dernière remarque se rapporte à l'étude des incitation à produire et à innover. Celles-ci doivent être analysées en relation avec le cadre institutionnel dans lequel elles s'inscrivent. Pour aller au delà de ce premier aperçu, il faudrait développer une approche plus approfondie des phénomènes cognitifs caractéristiques des deux systèmes de production qui ont été évoqués, en identifiant précisément les externalités économiques que l'on peut y associer. On pourrait plus spécifiquement qualifier la nature des incitations à innover et leur ampleur, ainsi que les formes de répartition de la valeur ajoutée créée, afin d'en déduire les possibilités de développement, la portée et la limite de l'un et l'autre des systèmes, leur stabilité, et éventuellement, l'intérêt économique de leur coévolution, si celle-ci s'avérait souhaitable.

Bibliographie

Arrow K. [1962]; "Economic Welfare and the Allocation of Resources for Invention" ; in Nelson R. (ed.) *The Rate and Direction of Inventive Activity : Economic andd Social Factors*, Princeton University press, N.J.

Behlendrof B. [1999]; "Open Sources Softwares" in DiBona C., Ockman S., Stone M. (eds.).

Bersini H., Kuntz P., Syers D., [1994]; "Algorithmes et intelligence collective" ch7 de Bonabeau E., Theraulaz G. (eds.) *Intelligence collective* ; Hermès, Paris p. 203-224.

Delapierre M., Gerard-Varet L.A., Zimmermann J.B. [1980]; "Choix Publics et normalisation des réseaux informatiques". Rapport BNI; décembre 1980.

DiBona C., Ockman S., Stone M. [1999]; *Open Sources, Voices from the Open Sources Revolution* ; O'Relly & Associates.

Di Cosmo R. & Nora D. [1998]; *Le hold-up planétaire*, Calmann-levy, Paris.

Foray D. [1995] ; "Les brevets dans la nouvelle économie de l'innovation" ; ch 6 de : Basle, M, Dufourt D., Heraud J.A., Perrin, J. (éds.), *Changement institutionnel et changement technologique Evaluation, droits de propriété intellectuelle, système national d'innovation*, CNRS Editions

Genthon C. [1995]; *Croissance et crise de l'industrie informatique*; Syros, Paris.

Genthon C. [1998]; Innovation et changement structurel: l'exemple de l'industrie informatique ; *Revue d'Economie Industrielle*, N° 85 3° trimestre 1998.

Haas S. [1995]; Dynamiques d'innovation : les enseignements de l'industrie des logiciels à Boston; *Terminal*, N° 69, automne 1995.

Jullien, Phan [1999]; From the "Unix World" towards the "Linux community : a historical co-evolutionary perspective on "small worlds""; paper to be presented at the 1999 EAEPE Conference, Prague, 4-7 November.

McKusic M.K. [1999]; Twenty Years of Berkeley Unix: From AT&T-Owned to Freely Redistributable in DiBona C., Ockman S., Stone M. (eds.).

Mowery D.C. (eds.) [1996]; *The International Computer Software Industry : a Comparative Study of Industry Evolution and Structure*; Oxford Univ Press.

Raymond E. [1998]; The Cathedral and the Bazaar ; http://www.tuxedo.org/~esr/writings/cathedral-bazaar/

version française (trad. Sébastien Blondeel): http://linux-france.com/article/these/cathedrale-bazar/

Rosenberg N. [1982]; *Inside the Black Box: Technology and Economics,* Cambridge University Press.

Salais R., Storper M. [1993]; *Les mondes de production* ; éditions de l'EHESS, Paris.

Schach S. [1990]; *Software Engineering*, Aksen Associates, Boston.

Smet-Sloanes J.P., Faucon B. [1999]; Logiciels Libres, Liberté, Egalité, Business. Edispher.

Stallman R.M. [1999] ; "The GNU Operating System and the Free Software Movement" ; in DiBona C., Ockman S., Stone M. (eds.).

Vixie, P. [1999]; in DiBona C., Ockman S., Stone M. (eds.).