

Les logiciels libres pour le gouvernement du Québec

Daniel Pascot

3 mars 2003

Professeur, Université Laval, Québec

Table des matières

1	Introduction	3
2	Qu'est-ce qu'un logiciel libre ?	4
2.1	Liberté et logiciel	4
2.2	Logiciels : les variantes du libre au propriétaire	7
2.3	Les logiciels libres disponibles	9
3	Au delà du logiciel libre, le phénomène du logiciel libre : un écosystème où règne la collaboration	10
3.1	Un phénomène complexe à aborder comme un écosystème	10
3.2	Un phénomène à dimension technique	13
3.3	Un phénomène à dimension humaine	14
3.4	Un phénomène à dimension organisationnelle	15
3.5	Un phénomène en perpétuelle évolution	17
4	Comment se fait-il que les logiciels libres existent et évoluent ?	19
4.1	Les conditions juridiques de la collaboration et de l'innovation	20
4.2	Les conditions techniques et l'innovation technologique	20
4.3	Le rôle des standards et normes : une vision historique de l'évolution vers le service	23
4.4	Les conditions socio-économiques	25
5	Pour une évolution vers l'usage des logiciels libres	29
5.1	Le logiciel est un bien immatériel	29
5.2	Un logiciel repose sur des normes	30
5.3	Un logiciel libre n'est pas (simplement) un logiciel gratuit	30

5.4	Le logiciel (qu'il soit libre ou propriétaire) n'est pas simplement un bien commercial que l'on acquiert	31
5.5	Éléments pour la construction de scénarios	34
5.5.1	Les ancrages en ce qui concerne la définition d'une politique face au logiciel libre : les raisons et motivations	34
5.5.2	Par où commencer	34
5.5.3	L'action gouvernementale	35
5.5.4	L'approche	36
5.5.5	Les écueils	36
6	En guise de conclusion : quatre questions, quatre réponses	36
6.0.6	Le Gouvernement du Québec doit-il envisager de participer au phénomène du logiciel libre : oui !	37
6.0.7	Le Gouvernement du Québec tirera-t-il des avantages à participer au phénomène du logiciel libre : oui !	37
6.0.8	Le Gouvernement du Québec a-t-il les moyens de participer au phénomène du logiciel libre : oui !	37
6.0.9	Le Gouvernement du Québec devrait envisager de participer au phénomène du logiciel libre : le plus tôt possible !	37
7	Annexes	38
7.1	La lettre du DR. EDGAR DAVID VILLANUEVA NUÑEZ, Membre du Congrès de la République du Pérou en réponse au Directeur Général de Microsoft Pérou	38
7.2	Témoignage	50

1 Introduction

Ce texte a été rédigé en conclusion d’une étude commandée par la Direction de l’architecture et de l’infrastructure du Sous-secrétariat à l’infrastructure gouvernementale et aux ressources informationnelles du Secrétariat du Conseil du trésor du gouvernement du Québec sur le phénomène des logiciels libres dans une perspective gouvernementale. Cette étude a renforcé l’intérêt de l’auteur de ce texte pour les logiciels libres (c’est en raison de son intérêt initial que l’étude lui a été confiée) ; en effet, elle lui semble confirmer que la réponse aux questions suivantes est positive :

- À la question “Le Gouvernement du Québec doit-il envisager de participer au phénomène du logiciel libre ?”, la réponse est oui !
- À la question “Le Gouvernement du Québec tirera-t-il des avantages à participer au phénomène du logiciel libre ?”, la réponse est oui !
- À la question “Le Gouvernement du Québec a-t-il les moyens de participer au phénomène du logiciel libre ?”, la réponse est oui !
- À la question “Quand le Gouvernement du Québec devrait-il envisager de participer au phénomène du logiciel libre ?”, la réponse est le plus tôt possible.

Face à ce sujet qui entraîne des débats parfois passionnés, l’auteur de document ne fait donc pas semblant d’être “neutre” et annonce ses couleurs¹, par contre il s’efforce d’être honnête et rigoureux comme le demande son statut de professeur d’université.

La formulation de ces questions est délibérée et illustre la position défendue dans ce rapport : il est proposé de participer au phénomène du logiciel libre et non pas simplement d’utiliser un logiciel libre : en effet, les logiciels libres ne sont pas le résultat d’une création spontanée, ils sont le résultat de la collaboration d’une communauté d’utilisateurs soucieux de la pérennité et de la sécurité des documents qu’ils créent avec leurs logiciels, cette communauté regroupe développeurs et utilisateurs.

La première partie définit ce qu’est un logiciel libre en terme technique et juridique ; la liberté est définie par rapport à l’accès au code source et est matérialisée par une licence, en particulier la licence GPL qui est probablement la plus intéressante pour un gouvernement. La deuxième partie tente de montrer que le logiciel libre ne devrait pas être considéré seulement comme un produit mais comme un phénomène complexe, un écosystème où règne la collaboration. De multiples dimensions doivent être prises en compte : technique bien sûr, mais aussi humaines et organisationnelles. De plus c’est un phénomène qui évolue rapidement. Ainsi l’économie du coût de la licence du logiciel devient un élément parmi d’autres. Lorsque l’on aborde la problématique des logiciels libres, une question émerge immédiatement : comment peut-on penser disposer gratuitement légalement d’un tel bien ? C’est ce à quoi tente de répondre la partie suivante. Le cadre juridique de la licence GPL est certainement nécessaire pour garantir la collaboration, mais cette collaboration repose aussi sur le mode de fonctionnement que les programmeurs ont développé grâce à Internet, ainsi que sur des normes et des standards. Enfin on explore la structure du “marché” du logiciel libre en contraste avec la structure du marché du logiciel propriétaire et on l’illustre sur deux situations, les logiciels de “e-learning” pour les universités et les logiciels de gestion de

¹On trouve à la fin de ce document, sous la forme d’un témoignage, la description de ses relations concrètes avec le monde des logiciels libres.

cabinets médicaux dans une perspective de communication dans le réseau de la santé, pour conclure que le recours aux logiciels libres n'est pas une utopie d'anarchiste mais une saine option de gestion.

On poursuit en faisant ressortir quelques éléments fondamentaux pour la définition d'une politique gouvernementale face aux logiciels libres :

- un logiciel est un bien immatériel (non rival, coût marginal nul, investissement faible, effet réseau), ceci a d'importantes conséquences,
- un logiciel repose sur des normes,
- un logiciel libre n'est pas simplement gratuit,
- un logiciel n'est pas simplement un bien commercial qu'on achète, c'est un moyen de communiquer et de conserver des informations

On conclut en énonçant les principaux éléments nécessaires pour construire des scénarios de participation au phénomène des logiciels libres, en précisant les arguments supportant les réponses aux quatre questions énoncées précédemment.

On a choisi d'adjoindre à ce document quatre annexes, l'une est une lettre du DR. EDGAR DAVID VILLANUEVA NUÑEZ, Membre du Congrès de la République du Pérou en réponse au Directeur Général de Microsoft Pérou. Parmi les documents explicitant la position d'un gouvernement face aux logiciels libres, cette lettre est exemplaire. Elle a fait le tour du monde (elle est traduite en plusieurs langues par des personnes promouvant les logiciels libres dans leur environnement), elle répond point par point aux principaux arguments des entreprises de logiciel propriétaire dont Microsoft est de toute évidence le principal représentant. Le deuxième rassemble deux décrets concernant l'utilisation des standards et des logiciels libres par l'Assemblée communautaire francophone de la ville de Bruxelles, ils synthétisent parfaitement l'engagement que peut prendre une administration publique. Le troisième document est le témoignage de l'auteur de ce document en ce qui concerne son usage des logiciels libres, il permet de prendre conscience du niveau atteint par les logiciels libres en ce qui concerne un usage personnel (bureautique). Enfin on présente un résumé des différents rapports constituant cette étude.

2 Qu'est-ce qu'un logiciel libre ?

2.1 Liberté et logiciel

Au cours de cette étude, qui a comporté une collecte de données sur le terrain mais aussi une collecte d'information dans la littérature, principalement celle qui est accessible dans Internet², nous avons été saisi par la dissonance entre d'une part le manque de connaissance de ce phénomène par bien des personnes rencontrées³ (et ceci même

²Ce qui est légitime car actuellement on ne peut pas dissocier le phénomène des logiciels libres d'Internet. On peut même dire que sans les logiciels libres, Internet ne serait pas ce qu'il est aujourd'hui et que sans Internet les logiciels libres n'auraient certainement pas pu prendre un tel essor.

³Ceci n'est pas propre au Québec. Dans une entrevue (qui remonte toutefois à 1998, les choses ont pu quelque peu changer depuis), Stéphane Fermigier disait que la diffusion des logiciels libres se heurtait à : 1. L'ignorance dans laquelle se trouvent les dirigeants d'entreprises à son sujet. 2. La crainte de sauter le pas, entretenue par des mythes inexacts (FUD – Fear, Uncertainty, and Doubt) : il n'y a pas de service d'assistance technique, il évolue trop vite, il convient aux bricoleurs et non aux utilisateurs sérieux

dans notre environnement universitaire direct de professeurs, d'étudiants ou d'administrateurs) et d'autre part par l'abondance d'information, de logiciels et d'aides relatifs aux logiciels libres disponibles ou identifiables dans Internet. Il convient donc, avant tout développement, de préciser ce que l'on entend par logiciel libre. En effet, bien que le logiciel libre existe depuis la création des premiers ordinateurs⁴ il reste méconnu ou incompris de la plupart des informaticiens, des gestionnaires et du grand public. Nous supposons que le lecteur sait ce qu'est un logiciel (que ce soit un système d'exploitation, un outil de développement ou une application plus ou moins spécialisée), il reste à préciser ce que l'on entend par "libre". Le mieux est de se référer aux sources et en particulier à la FSF "Free Software Foundation" ou "Fondation pour le logiciel libre"^{5,6}, qui, bien qu'elle ne soit pas le seul acteur du phénomène du logiciel libre, et qu'elle soit parfois contestée pour son purisme, en particulier par les tenants du mouvement "open source"^{7,8}, elle n'en demeure pas moins la référence obligée en ce qui concerne le concept de logiciel libre :

L'expression « Logiciel libre » fait référence à la liberté et non pas au prix. Pour comprendre le concept, vous devez penser à la « liberté d'expression », pas à « l'entrée libre ». Un logiciel libre est un logiciel qui est fourni avec l'autorisation pour quiconque de l'utiliser, de le copier, et de le distribuer, soit sous une forme conforme à l'original, soit avec des modifications, ou encore gratuitement ou contre un certain montant. Ceci signifie en particulier que son code source doit être disponible. « S'il n'y a pas de sources, ce n'est pas du logiciel libre. »⁹

La définition du logiciel libre est donc directement associée à la disponibilité du code source¹⁰. En effet, on ne peut normalement pas faire tout ce que l'on désire avec un logiciel (personnellement ou avec l'aide d'une autre personne, éventuellement payée pour le faire) si l'on ne dispose que de la version exécutable. Plus précisément ces libertés sont :

(lire <http://www.smets.com/it/faq/linux.html>) 3. La confusion entre gratuité et libre disponibilité des sources, deux notions souvent rendues par le même terme, freeware ou free software, et qui recouvrent des notions complètement différentes. 4. Des facteurs pervers liés au mode de gestion des entreprises : il n'est pas sérieux pour un DSI de réclamer un budget de 250 F pour l'achat des logiciels systèmes et d'une partie des applications dont il a la responsabilité. <http://www.iful.org/faqs/interview.html>

⁴Dans le très intéressant ouvrage "Tribune libre" qui rassemble les témoignages des principaux acteurs du monde des logiciels libres <http://www.editions-oreilly.fr/divers/tribune-libre/index.html>, le premier chapitre, "Une brève histoire des hackers", précise que l'on peut placer le point de départ de la culture des hackers (les "vrais programmeurs" auteurs des logiciels libres), telle qu'on la connaît, en 1961, l'année où le MIT a fait l'acquisition du premier PDP-1.

⁵<http://www.gnu.org>

⁶Pour le chapitre français de cette fondation, voir : <http://france.fsfeurope.org/>

⁷<http://www.opensource.org>

⁸Pour une explication de la différence, somme toute mineure, entre Open source et Free Software voir <http://www.debian.org/intro/free> et certains développements relatifs aux aspects juridiques et économiques des licences de logiciel.

⁹<http://www.gnu.org/philosophy/categories.fr.html>

¹⁰Précisons que le code source est le programme (un ensemble de textes composés chacun d'une suite d'instructions) écrit par le programmeur alors que le code objet est la traduction (automatique) du programme de telle sorte qu'il soit exécutable par un ordinateur doté du système d'exploitation approprié : le code objet est hermétique et ne peut que très difficilement être modifié sans risque élevé de mauvais fonctionnement.

- La liberté d'exécuter le programme, pour tous les usages (liberté 0).
- La liberté d'étudier le fonctionnement du programme, et de l'adapter à vos besoins (liberté 1).
- La liberté de redistribuer des copies, donc d'aider votre voisin, (liberté 2).
- La liberté d'améliorer le programme et de publier vos améliorations, pour en faire profiter toute la communauté (liberté 3).

Un programme est un logiciel libre si les utilisateurs ont toutes ces libertés. Ainsi, vous êtes libre de redistribuer des copies, avec ou sans modification, gratuitement ou non, à tout le monde, partout. Être libre de faire ceci signifie (entre autre) que vous n'avez pas à demander ou à payer pour en avoir la permission.

...

La liberté d'utiliser un programme est la liberté pour tout type de personne ou d'organisation de l'utiliser pour tout type de système informatique¹¹, pour tout type de tâche et sans être obligé de communiquer ultérieurement avec le développeur ou tout autre entité spécifique.

...

Pour que ces libertés soient réelles, elles doivent être irrévocables tant que vous n'avez rien fait de mal ; si le développeur du logiciel a le droit de révoquer la licence sans que vous n'avez fait quoi que ce soit pour le justifier, le logiciel n'est pas libre.

...

Ainsi, vous pouvez avoir à payer pour obtenir une copie d'un logiciel du projet GNU¹² ou vous pouvez l'obtenir gratuitement. Mais indifféremment de la manière dont vous vous l'êtes procuré, vous avez toujours la liberté de copier et de modifier un logiciel et même d'en vendre des copies.

« Logiciel libre » ne signifie pas « non commercial ». Un logiciel libre doit être disponible pour un usage commercial. Le développement commercial de logiciel libre n'est plus l'exception ; de tels programmes sont des logiciels commerciaux libres¹³.

Tout le monde n'interprète pas de la même manière ces quatre libertés qui sont formalisées dans le cadre d'une licence, la licence GPL (General Public License)¹⁴, laquelle protège de nombreux et importants logiciels libres. Mais ce n'est pas la seule licence

¹¹Pour transporter un programme d'un environnement informatique à un autre il faut disposer du code source (besoin de recompilation).

¹²Le projet GNU (le nom est basé sur le jeu de mot récursif : GNU is not Unix et non sur l'animal -le gnou- qui en est devenu le symbole) est un projet fondamental indissociablement lié à la FSF. Il consiste à développer et fournir les outils de développement libres, incluant un système d'exploitation, indispensables à la réalisation de tout programme. Sans les logiciels du projet GNU, Linux n'existerait pas ; c'est pour cela que selon Richard M. Stallman, le père du projet GNU et de la FSF ainsi que de la GPL (General Public Licence), on devrait dire GNU/Linux.

¹³<http://www.gnu.org/philosophy/free-sw.fr.html>

¹⁴Voir w.april.org/gnu/gpl_french.html pour une traduction non-officielle de la licence GPL (seule la version anglaise pour des raisons d'économie et de simplification juridique est actuellement considérée comme officielle par la FSF).

disponible¹⁵, il en existe plus d'une trentaine et rien n'interdit d'en créer d'autres. Les plus connues sont la licence artistique¹⁶, la licence BSD¹⁷, ou des licences écrites pour répondre à une situation particulière comme la licence Netscape ou Mozilla pour le futur du même nom¹⁸, mais elles participent toutes d'une même philosophie de liberté servant de garantie à la collaboration.

2.2 Logiciels : les variantes du libre au propriétaire

Cette notion de logiciel libre s'oppose à la notion de logiciel propriétaire. Un logiciel dit propriétaire est la propriété de quelqu'un (personne physique ou morale) qui en restreint les droits d'usage tout en limitant sa responsabilité à l'aide d'une licence (essentiellement usage limité à un seul poste de travail et non responsabilité des conséquences néfastes de l'usage du logiciel) afin de procurer des revenus à son propriétaire, ce qui amène souvent (mais pas obligatoirement) à ne pas divulguer le code source. Cependant la situation est souvent confuse, pour bien fixer les idées voici les principales définitions des classes de logiciel autres que le logiciel libre telles que présentées par la FSF :

- **Logiciel Open Source** : Le terme logiciel « open source » (littéralement à source ouvert) est utilisé par certaines personnes pour signifier plus ou moins la même chose que logiciel libre tel que défini par la FSF.

- **Logiciel du domaine public** : Logiciel du domaine public veut dire logiciel non soumis aux droits d'auteurs. C'est un cas particulier de logiciel libre "non-copylefté", ce qui veut dire que certaines copies, ou certaines versions modifiées, peuvent ne pas être du tout libres. Parfois, on utilise le terme « domaine public » d'une façon peu précise pour dire « libre » ou « disponible gratuitement ». Toutefois, « domaine public » est un terme légal qui signifie précisément que le logiciel n'est pas « soumis au copyright ». Afin d'être plus précis, nous conseillons d'utiliser le terme « domaine public » dans ce cas uniquement, et d'utiliser d'autres termes dans les autres cas.

- **Logiciel copylefté (sous gauche d'auteur)** : Le logiciel sous copyleft (littéralement, gauche d'auteur) est un logiciel libre, dont les conditions de distribution interdisent aux nouveaux distributeurs d'ajouter des restrictions supplémentaires lorsqu'ils redistribuent ou modifient le logiciel. Ceci veut dire que chaque copie du logiciel, même si elle a été modifiée, doit être un logiciel libre. Le copyleft est un concept général ; pour l'appliquer à un programme, vous avez besoin d'un ensemble de termes re-

¹⁵On trouve sur le site de l'AFUL (L'AFUL est une association d'utilisateurs et de professionnels du logiciel libre. Elle comprend des membres - individus, sociétés commerciales, autres associations - dans une dizaine de pays ou régions du monde francophones) des liens procurant une information assez complète sur les licences <http://www.aful.org/presentations/licences/index.html>

¹⁶Voir par exemple la Licence Art Libre <http://artlibre.org/faq.php/definitions.html> ou encore la Licence ludique générale <http://jeuxlibres.free.fr/llg.htm> ou encore <http://www.linux-france.org/article/these/licence/artistic/fr-artistic.html>

¹⁷<http://www.freebsd.org/copyright/license.html>

¹⁸<http://www.mozilla.org/MPL/>

latifs à sa distribution. Il y a de nombreuses façons d'écrire ces conditions de distribution.

- **Logiciel libre non-copylefté** : Le logiciel libre non-copylefté est diffusé par son auteur avec la permission de le redistribuer et de le modifier, mais aussi d'y ajouter d'autres restrictions. Si un programme est libre, mais non-copylefté, alors certaines copies ou versions modifiées peuvent ne plus être libres du tout (comme avec la licence BSD). Une société informatique peut compiler ce programme, avec ou sans modifications, et distribuer le fichier exécutable sous forme de produit logiciel propriétaire.

- **Logiciel semi-libre** : Le logiciel semi-libre est un logiciel qui n'est pas libre, mais qui s'accompagne de la permission pour les personnes physiques de l'utiliser, de le copier, de le distribuer, et de le modifier (y compris pour la distribution des versions modifiées) dans un but non lucratif. PGP est un exemple de programme semi-libre. Un logiciel semi-libre est toujours mieux qu'un logiciel propriétaire, mais cela pose toujours des problèmes, et nous ne pouvons l'utiliser dans un système d'exploitation libre. Les restrictions du copyleft sont conçues pour protéger les libertés fondamentales pour tous les utilisateurs. Pour nous, la seule justification à la définition d'une restriction substantielle sur l'utilisation d'un programme est d'empêcher l'ajout d'autres restrictions par d'autres personnes. Les programmes semi-libres possèdent des restrictions supplémentaires, motivées par des buts purement égoïstes.

- **Logiciel propriétaire** : Le logiciel propriétaire est un logiciel qui n'est ni libre, ni semi-libre. Son utilisation, sa redistribution ou sa modification sont interdites, ou exigent une autorisation spécifique, ou sont tellement restreintes que vous ne pouvez en fait pas le faire librement.

- **Freeware** : Le terme « freeware » n'a pas de définition claire communément acceptée, mais il est utilisé couramment pour des paquets qui autorisent la redistribution mais pas la modification (et dont le code source n'est pas disponible). Ces paquets ne sont pas des logiciels libres, donc n'utilisez pas, s'il vous plaît, « freeware » pour parler de logiciel libre.

- **Shareware (partagiciel)** : Le partagiciel est un logiciel qui s'accompagne de la permission de redistribuer des copies, mais qui mentionne que toute personne qui continue à en utiliser une copie est obligée de payer des royalties. Les sharewares ne sont pas des logiciels libres ou même semi-libres. Pour deux raisons : pour les sharewares, le code source n'est pratiquement jamais fourni ; et donc vous ne pouvez pas du tout modifier le programme ; avec le shareware, il ne vous est pas permis d'effectuer une copie et de l'installer sans vous acquitter du paiement d'un droit licence, même pour des individus impliqués dans des activités non lucratives. (En pratique, ces termes de distribution sont en général peu appréciés, et les gens le font quand même, même si ce n'est pas permis.)

- **Logiciel commercial** : Le logiciel commercial est un logiciel développé par une entreprise dont le but est de gagner de l'argent sur l'utilisation du logiciel. « Commercial » et « propriétaire » ne sont pas synonymes ! La plupart des logiciels commerciaux sont propriétaires, mais il y

a des logiciels libres commerciaux, et il y a des logiciels non-commerciaux non-libres. Par exemple, GNU Ada est toujours distribué sous les termes de la GPL GNU, et chaque copie est un logiciel libre ; mais ses développeurs vendent des contrats de support. Quand leurs commerciaux parlent à de futurs clients, quelquefois ceux-ci disent, « Nous nous sentirions plus en sécurité avec un compilateur commercial. ». Le représentant répond, « GNU Ada est un compilateur commercial ; il est également un logiciel libre. » Pour le Projet GNU, l'accent est mis sur l'autre composante : la chose importante est que GNU Ada est un logiciel libre ; que ce soit un logiciel commercial n'est pas un point crucial. Cependant, le développement supplémentaire de GNU Ada qui résulte de ce commerce est certainement bénéfique.¹⁹

Ainsi la caractéristique fondamentale du logiciel libre est la liberté d'accès au code source qui garantit la liberté d'usage et non pas le caractère commercial du logiciel. Cette dimension se trouve au coeur du mouvement dit "open source" qui sur certains points se démarque du mouvement "free software"²⁰. La polémique est loin d'être éteinte entre les deux mouvements que l'on peut considérer comme frères, mais tout de même un peu ennemis ; cependant leur différent est loin d'en être au point qu'ils ne fassent pas corps face aux logiciels propriétaires, c'est ainsi que l'on rencontre de plus en plus souvent l'expression FOSS (Free Open Source Software).

La dimension juridique est fondamentale dans la mesure où la coopération dans un univers économique a besoin d'un encadrement et d'une protection juridique, surtout en Amérique du Nord où se trouve la plus grande population d'avocats au monde. La dimension monétaire importe aussi, mais elle n'est pas la seule. C'est cette liberté d'accès au code source qui permet la coopération et qui fait que le phénomène du logiciel libre existe. La dimension juridique du logiciel libre est une dimension d'autant plus complexe qu'elle se déploie dans un contexte international, mais elle est loin à elle seule de permettre d'appréhender le phénomène.²¹ Le fait que les programmes soient dépendants l'un de l'autre (tout programme a besoin d'autres programmes pour pouvoir effectivement fonctionner) et doivent évoluer rend la dimension juridique très complexe car toutes les licences ne sont pas compatibles entre elles et n'autorisent pas les modifications de la même manière : le risque de fouillis juridique est loin d'être nul si aucune attention n'est donnée à cette dimension.

2.3 Les logiciels libres disponibles

Aujourd'hui de nombreux besoins peuvent être comblés par des logiciels libres. Le principal représentant des logiciels libres, celui qui s'impose en premier, le seul que l'on connaisse (de nom) si l'on en connaît qu'un est Linux, un système d'exploitation complet, puissant, robuste qui supporte presque la moitié des serveurs d'Internet (avec d'autres variantes libres de Unix comme freeBSD) et qui est en passe de devenir le

¹⁹<http://www.gnu.org/philosophy/categories.fr.html>

²⁰<http://www.gnu.org/philosophy/categories.fr.html>

²¹Parmi bien d'autres, l'article de Valérie Sédallian "Garanties et responsabilités dans les logiciels libres" illustre bien la complexité juridique du monde du logiciel qu'il soit libre ou pas. <http://www.juriscom.net/pro/2/da20020901.htm>

choix principal d'IBM et de nombreux grands utilisateurs comme le gouvernement Allemand (essentiellement pour des raisons de sécurité). Linux est par exemple reconnu le choix numéro un pour constituer un parefeu.

Une application intranet peut être développée entièrement avec des logiciels libres, les programmeurs disposent de compilateurs pour tous les langages connus et pratiqués (gcc est considéré comme le meilleur compilateur C au monde), d'éditeurs très puissants (emacs, vi), d'environnement de gestion des sources (CVS), de systèmes de gestion de bases de données (MySQL, Postgress), d'environnements de programmation et de serveurs d'applications (Apache, Tomcat, Struts, ... du projet Jakarta²²). On a formé l'acronyme LAMP : Linux (système d'exploitation et utilitaires), Apache (serveur d'application), MySQL (base de données relationnelle) et PHP (langage de programmation d'applications Web) pour désigner une solution maintenant classique pour le développement des applications Web, que l'on retrouve dans les coquilles comme Zope, Spip, ... bref on commence à avoir l'embarras du choix !

Pour ce qui est de l'accès à Internet, certains disent même que Mozilla, la version libre de Netscape est actuellement en avance sur Internet Explorer de Microsoft²³ et ce n'est pas le seul logiciel disponible en tant que fureteur, courrier électronique, transfert de fichier, clavardage et autres besoins liés aux échanges.

En ce qui concerne la bureautique les solutions ont émergé et sont de plus en plus près de rivaliser avec les meilleures solutions propriétaires, de plus elles sont en avance en ce qui concerne XML. Le problème n'est plus dans les fonctionnalités, mais les compatibilités de format de document. Les environnements de bureau comme KDE et Gnome ont atteint une qualité et un niveau de sophistication tel qu'il devient raisonnable de les envisager dans un usage professionnel quotidien.

De très nombreux besoins peuvent être satisfaits par des logiciels libres depuis les besoins fondamentaux, en passant par le graphisme ou la composition de documents ou l'édition musicale jusqu'aux jeux (on a l'embarras du choix pour faire une réussite ... !) et au visionnement des films vidéos de tous formats.

3 Au delà du logiciel libre, le phénomène du logiciel libre : un écosystème où règne la collaboration

3.1 Un phénomène complexe à aborder comme un écosystème

Le phénomène des logiciels libres est un phénomène complexe présentant de multiples facettes, qui, malgré son âge et son ampleur, peut encore être considéré comme un monde mystérieux tant il est encore inconnu ou mal compris par de nombreux informaticiens ou la plupart des gestionnaires utilisateurs de l'informatique, et en général du grand public. Dès que l'on tente de cerner ce phénomène on est frappé par ses aspects paradoxaux et contre-intuitifs²⁴. Pour de multiples raisons c'est un univers en-

²²<http://jakarta.apache.org/>

²³101 trucs que Mozilla sait faire et qu'IE ne sait pas http://clx.anet.fr/spip/article.php3?id_article=158, évidemment il se peut que IE en fasse quelques uns que Mozilla ne fait pas.

²⁴Dans sa thèse de doctorat en sciences économique François Horn parle d'un monde en contraste et d'un paradoxe déconcertant pour un économiste http://rntl.alcove.com/documents_universitaires/

core secret : nous avons rencontré des programmeurs qui utilisent des logiciels libres en cachette de leurs supérieurs, et si l'on fait une recherche dans Internet au sujet des universités québécoises avec le mot clé Linux ou logiciel libre on ne trouve presque rien, alors que certaines personnes y tentent et parviennent de plus en plus à n'utiliser que des logiciels libres et en font régulièrement la promotion.

L'étude du phénomène du logiciel libre devrait être abordée comme celle d'un écosystème²⁵, en effet ce phénomène est le résultat (en perpétuel changement) de facteurs appartenant à de multiples dimensions : technique (ingénierie, progrès, performance, qualité, standards et normes, ...), humaine (motivation, psychologie, résistance au changement, formation, attitudes ou croyances, ...) organisationnelle (organisation du travail, gestion de projet, législation, économie, ...). Il doit être interprété à la lumière de concepts de nature individuelle (portée locale), mais aussi souvent en même temps de concepts de nature collective (portée globale). C'est pour cette raison que nous parlons ici du "phénomène des logiciels libres" et non pas seulement de "logiciels libres".

La "valeur" des logiciels quels qu'ils soient n'est que très partiellement reflétée par leur valeur marchande²⁶, car leur "valeur d'usage" repose sur leur création, réalisation, mise en oeuvre et utilisation. Cette "valeur d'usage" comporte de multiples dimensions, elle devrait être abordée dans une perspective systémique multidimensionnelle et évolutive comme le font les chercheurs en écologie²⁷. En effet, le phénomène des logiciels libres ne peut pas être limité à une dimension commerciale (ou marchande). Par exemple la firme de recherche commerciale IDC estime que les revenus de Linux en 2001 ont chuté de 5% par rapport à ceux de l'année précédente ; ils auraient été de 80 millions \$ US, ce qui représente moins de 0,05% des ventes de système d'exploitation. Cette performance commerciale est en complète contradiction avec la reconnaissance de plus en plus grande accordée à son créateur initial, Linus Torvald²⁸, avec le nombre de documents accessibles par Internet ou disponibles en librairie (le rayon des livres consacrés à Linux dans les librairies spécialisées en informatique de la plupart des pays du monde est aussi fourni sinon plus que le rayon consacré aux différentes versions de Windows), ou encore avec le nombre d'études consacrées à la comparaison entre Linux et Windows.

L'importance du phénomène des logiciels libres ne peut pas non plus être cernée simplement par la dimension économique attachée à un logiciel et mesurée par exemple par le TCO ("Total Cost of Ownership" : coût total de possession). D'une part il apparaît qu'en moyenne le coût d'acquisition d'un logiciel ne représente qu'une petite partie des coûts directement mesurables (souvent moins de 10%) et, d'autre part, cela

[these_horn/introduction.pdf](#)

²⁵J'ai emprunté cette expression à Hilaire Fernades dans son papier "La classe, le gnou et le pingouin" <http://www.ofset.org/information/papers/free-software-and-education-fr.html>

²⁶Dans son article emblématique "The magic cauldron" Eric S. Raymond (<http://adonnart.free.fr/doc/magichau.htm> pour une traduction en Français) présente la différence entre la valeur d'acquisition ou de vente et la valeur d'utilisation : la valeur d'utilisation est la valeur en tant qu'outil alors que la valeur d'acquisition est la valeur sur le marché.

²⁷On lira avec intérêt le document Tribune libre qui rassemble les témoignages des auteurs les plus connus et importants du monde des logiciels libres <http://www.editions-oreilly.fr/divers/tribune-libre/index.html>, il permet de percevoir grâce à sa dimension historique (qu'elle soit technologique, économique ou juridique) l'essence du mouvement du libre dans sa diversité.

²⁸Certaines reconnaissances peuvent être surprenantes comme celle de l'homme de l'année (1999) dans la catégorie commerce <http://www.wtn.net/awards/awards2001/commerce/commerce.html>

supposerait un comportement irrationnel de la part des décideurs : pourquoi choisir si souvent la solution la plus coûteuse ?

L'importance du phénomène ne se résume pas non plus à un problème technique, car il est largement reconnu que la qualité technique de nombreux logiciels libres est au moins aussi bonne que celle des logiciels propriétaires et que l'assistance que l'on peut obtenir est comparable, sinon de meilleure qualité²⁹.

Ce n'est pas non plus simplement, comme le clament certains, une réaction de "communistes"³⁰ face au tout puissant monopole de Microsoft, car les logiciels libres existent en fait depuis le début de l'informatique, donc bien avant la création de Microsoft³¹³²³³.

L'usage d'un logiciel libre ne résulte pas simplement de sa disponibilité. Comment peut-on alors expliquer le non usage de certains logiciels libres au profit de logiciels propriétaires plus difficile d'accès et de surcroît onéreux ?

²⁹Par exemple voir la comparaison entre les différents Unix, Linux et Windows NT de John Kirch, dans sa traduction en français : <http://www.linux-france.org/article/these/unix-vs-nt/unix-vs-nt.html>

³⁰Il faut bien avouer que la barbe et la chevelure de Richard M Stallman le fondateur de la FSF et ses engagements politiques peuvent en effaroucher certains <http://www.stallman.org/>

³¹La compagnie Microsoft est à la base (ou la cible) de bien des réactions visant à favoriser (ou à contrecarrer) la progression du phénomène des logiciels libres. On en trouve de nombreuses manifestations sur Internet, certaines bien articulées, d'autres plus ou moins folkloriques, comme en témoigne le site <http://www.wehavethewayout.com>, signé par Microsoft et Unisys, lancé un premier avril (voir l'article http://it.mondelinux.org/article.php3?id_article=127) auquel a immédiatement répondu le site <http://www.wehavethewayin.com/section.php/about.htm> (actuellement ce site renvoie à <http://www.linuxfreak.org/>). Le site [wehavethewayout](http://www.wehavethewayout.com) redirige (encore actif en octobre 2002) vers <http://www.microsoft.com/windows2000/migrate/unix/whymigrate.asp?rew=0> afin d'être informé au sujet de "Migrating to Windows from UNIX and Linux" ... Parfois on se demande où est la farce, et pourtant comme le rapporte l'article cité ci-dessus : "Si ce site Internet, lancé le 1er avril, est un paradoxe, ce n'est pas un poisson d'avril : nous nous en sommes assurés au vu d'annonces antérieures évoquant le lancement de cette campagne. D'ailleurs, il aurait été moins catastrophique pour les deux sociétés que ce soit un canular... ", cela vaut mieux pour une campagne de 25 millions de dollars. Pour la petite histoire voici le début de l'article : "Microsoft et le groupe informatique américain Unisys ont lancé lundi 1er avril un site censé persuader des entreprises de migrer du système d'exploitation Unix vers Windows / Microsoft et le groupe informatique américain Unisys ont lancé lundi 1er avril un site censé persuader des entreprises de migrer du système d'exploitation Unix vers Windows. Baptisé "Wehavethewayout.com" ("Nous avons la solution" - way out signifiant également "migration"), ce site est pourtant un véritable paradoxe. / Selon Netcraft, société américaine qui détermine la nature des logiciels utilisés par les sites Internet, les serveurs de Wehavethewayout.com tournent sous FreeBSD, une version non propriétaire d'Unix, avec Apache comme logiciel d'administration web. Un comble qui a vite suscité la colère, ou la risée, des programmeurs orientés vers les systèmes ouverts (open source). / Retournement de situation : mardi 2 avril à midi, ce site de promotion des "solutions" Windows n'était plus accessible, remplacé par un écran blanc avec la mention "Erreur 404"".

³²Par ailleurs la lecture des documents dits Halloween [<http://www.opensource.org/halloween/halloween7.php>] semblent révéler que d'une part que Linux et le mouvement des logiciels libres sont considérés par Microsoft comme des concurrents sérieux sinon potentiels et d'autre part permettent de comprendre pourquoi des firmes comme Sun ou la défunte Be continuent à attaquer Microsoft en justice pour pratique monopolistique anticoncurrentielle.

³³Voir aussi l'article de Tim O'Reilly "The Strange Case of the Disappearing Open Source Vendors" <http://www.oreillynet.com/pub/a/network/2002/06/28/vendor.html>

3.2 Un phénomène à dimension technique

Un logiciel est un objet de nature technologique. C'est ainsi que l'on parle dans tous les milieux des NTIC (Nouvelles Technologies de l'Information et des Communications) et des "e-commerce", "e-gouvernement", "e-éducation" et bien d'autres acronymes et néologismes. La technologie informatique est en effet utilisée dans de plus en plus de domaines au fur et à mesure de son évolution. En particulier elle a envahi le domaine des communications, qu'elles soient écrites ou orales ; on écrit et publie de moins en moins sans l'aide d'un ordinateur ; on ne parle plus à distance sans l'aide de l'informatique (sans ordinateur, pas de téléphone cellulaire) ; on ne voyage plus sans l'aide d'ordinateurs (aussi bien dans les véhicules pour le contrôle du moteur ou la navigation, que pour les réservations ou la gestion du trafic). Toutes les technologies de création, conservation, communication et traitement de l'information sont maintenant visées par l'informatique. On trouve, par exemple, maintenant chez tous les fournisseurs de matériel de bureau des unités périphériques bon marché (pour quelques centaines de dollars) assurant les fonctions d'imprimante, de télécopieur, de photocopie et de scanner : le même instrument permet de manipuler toutes les formes de communication, orale, textuelle, image. On peut aujourd'hui écouter la radio ou un CD-audio, visionner un film reçu par le même réseau de distribution que les documents de commerce électronique, sur l'ordinateur qui sert de poste de travail dans un bureau typique. Tout ce qui est basé sur l'information est de plus en plus directement ou indirectement dépendant de l'informatique.

Or un ordinateur ne peut pas être utilisé directement. On doit utiliser des programmes pour toute manipulation ou conservation de l'information. Le logiciel est en fait, plus que la technologie de construction des ordinateurs et des matériels de télécommunication, ce sur quoi repose l'ensemble des NTIC. Bien que l'usage des logiciels nécessite de moins en moins de connaissances techniques, leur création en demande. Sans la connaissance technique de la construction du logiciel, il est impossible de faire autre chose que ce qui a été prévu par les concepteurs et permis par les réalisateurs des logiciels. Tout nouvel usage requiert de la programmation. Bien sûr il y a différents niveaux de programmation. Ainsi la mise en forme d'un texte n'exige, grâce aux logiciels disponibles, que des connaissances minimales, ceci tant que les fonctions du logiciels sont satisfaisantes et tant que l'on ne veut pas inter connecter des logiciels conçus indépendamment l'un de l'autre et que le système informatique que l'on utilise ne tombe en panne. La programmation prend de multiples formes : très technique et proche de la machine pour la réalisation des systèmes d'exploitation ou des logiciels de communication, plus abstraite, mais non moins technique pour la réalisation des logiciels de gestion, plus spécialisée pour la réalisation des sites Web (ou des jeux), mais dans tous les cas elle exige une connaissance spécialisée indépendante du domaine d'utilisation.

Il n'est donc pas étonnant de constater que les logiciels libres ont d'abord été développés par des programmeurs pour satisfaire leur propres besoins³⁴. Ces programmeurs communiquaient entre eux, souvent dans un environnement de recherche universitaire. Ils étaient des spécialistes plus ou moins concernés et contraints par les dimensions juridico-économiques des logiciels : leurs connaissances techniques les protégeaient

³⁴Compilateurs, éditeurs de texte -pour écrire les programmes-, bibliothèques de procédures et fonctions, utilitaires de gestion de fichiers, logiciels de gestion de télécommunications et de réseaux ...

de leur environnement et leur permettaient de définir leur plan de travail en fonction de leurs propres contraintes et intérêts. Leur autonomie ou indépendance, dans le contexte de leur travail, ou de leurs loisirs, les autorisait à partager leur travail sans contraintes monétaires. Les échanges entre les auteurs de logiciel (qu'il ne faut pas confondre avec les éditeurs qui sont les entreprises commerciales de vente de logiciels) est de nature technique. Grâce à Internet, leur réseau est maintenant mondial et ils ont pu ainsi accumuler une énorme base de connaissances. Ainsi si l'on a aujourd'hui un problème avec un logiciel propriétaire, une recherche dans Internet conduit presque inexorablement au site de service de l'éditeur du logiciel (quand il existe), alors qu'une recherche concernant un logiciel libre permet d'accéder aux documents de conception, à de la documentation et des échanges entre programmeurs et utilisateurs dans les diverses archives des forums de discussion. "Google" est ainsi devenu un outil quotidien essentiel à tout développeur, administrateur ou utilisateur de logiciel libre. Il permet avec quelques mots clés bien choisis de localiser quasi instantanément l'information requise si elle existe. C'est là une différence technique énorme entre le monde des logiciels propriétaires et le monde des logiciels libres.

3.3 Un phénomène à dimension humaine

Nous avons progressivement été conduits à identifier intuitivement³⁵ deux schèmes attitudeux face au phénomène du logiciel libre que l'on peut caricaturiser sous la forme de deux groupes ou « clans ». Nous utilisons le mot clan afin d'éviter le mot "religion" que l'on a parfois entendu ou lu. Actuellement (ceci peut évoluer) l'un de ces deux clans présente un intérêt marqué pour le phénomène du logiciel libre, l'autre présente de l'indifférence et bien souvent de l'hostilité. Le premier clan est celui des personnes qui sont conscientes des dimensions collectives ou sociales à moyen ou long terme de leurs choix et de leurs gestes. L'autre clan est celui des personnes qui privilégient la satisfaction de leurs besoins individuels à court terme et de leur confort. Ces clans ou attitudes apparaissent parfois crûment dans certaines situations ou certains forums de discussion au sujet des situations où il est question de l'adoption de logiciels libres.

La différence de comportement entre les membres des deux clans est frappante dans certains forums. Ainsi par exemple, la mise à disposition d'ordinateurs portables pour des élèves de la région des Landes a donné lieu à des débats intéressants entre les initiateurs et gestionnaires du projet et les promoteurs des logiciels libres. Un des responsables faisait valoir qu'il n'était pas intéressé par la provenance des logiciels mais qu'il voulait quelque chose qui réponde à ses propres critères sans lui poser de problème. Un des participants au débat (dans le forum du projet) lui faisait valoir qu'il avait la responsabilité de former des citoyens et non pas des consommateurs. Ainsi on peut penser que l'argument du manque d'une certaine fonctionnalité du coupé-collé dans OpenOffice donné comme l'une des raisons du rejet de ce logiciel ne vaut pas grand chose dans un contexte de logiciel libre. Que vaut-il par rapport à l'ensemble du projet, et pourquoi ne pas rentrer en contact avec les développeurs du logiciel ? Le

³⁵intuitivement : il ne s'agit pas ici d'une recherche empirique effectuée dans les règles de l'art, mais quelque chose qui s'est progressivement imposé à notre esprit.

problème serait probablement réglé avant bien longtemps, peut-être même avant que le logiciel n'arrive entre les mains des élèves. Il est intéressant de remarquer que cette dernière intervention n'a même pas été évoquée alors qu'elle est à la base de l'esprit des logiciels libres : la coopération et le contrôle de l'évolution par les utilisateurs. On voit bien tout au long des échanges dans ce projet l'opposition entre les deux attitudes et la difficulté de changer les mentalités de consommateurs face au logiciel que ce soit auprès des élèves ou de leurs parents et bien sûr des professeurs et des gestionnaires du projet³⁶.

Le besoin d'un "parapluie psychologique" pour les décideurs est caractéristique de l'une de ces attitudes. Ainsi va l'explication du refus d'un gestionnaire de mettre en place un serveur Linux basé sur l'argument suivant : "... et qu'est-ce que je fais si le serveur tombe en panne le vendredi soir ?". On ne peut que lui répondre "la même chose que si vos serveurs sont sous Windows, sauf que cela risque d'arriver moins souvent. Vous rappelez votre administrateur de système ou vous faites comme maintenant vous attendez à lundi ...". "Oui mais ... c'est moi qui sera alors responsable de ma décision alors que ...". Quiconque a un tant soit peu vécu dans les organisations, aussi bien publiques que privées, devine facilement la suite.

Ceci nous conduit à penser qu'au moins en ce qui concerne les logiciels, la disponibilité ne crée pas forcément la connaissance et l'usage de ces logiciels et que les attitudes et croyances ont un effet déterminant sur les comportements au même titre que les caractéristiques intrinsèques des outils. Ainsi, pour la direction d'une organisation qui, au constat des avantages potentiels du phénomène des logiciels libres, désire aller de l'avant, la diffusion de la connaissance et la prise en compte des attitudes sera un facteur critique de succès clé : il sera nécessaire de reconstituer le "parapluie psychologique" des décideurs.

3.4 Un phénomène à dimension organisationnelle

Un logiciel implique l'existence d'une communauté que ce soit pour sa création ou son utilisation. La complexité et la taille des logiciels est telle que même le meilleur et le plus productif des programmeurs ne peut pas compter uniquement sur lui-même, à moins de se restreindre à une fonction très limitée, il ne peut pas travailler seul et il a besoin de toutes façons d'utiliser les résultats du travail préalable de nombreux autres programmeurs qui ont réalisé les logiciels dont il a besoin pour produire son nouveau logiciel (on ne repart pas à zéro à chaque fois !). La production du logiciel est donc un domaine où le travail de groupe et l'échange entre les producteurs est fondamental : ce sont là les fondements de toute économie pour ne pas dire de toute société.

Il est rare qu'on utilise un logiciel pour satisfaire isolément ses propres besoins : la valeur d'une information est généralement définie par ce qu'elle apporte dans le contexte d'un échange. Ainsi un logiciel repose en grande partie sur la définition de contenus et de leurs usages dans un contexte d'échange au sein d'une communauté dont les membres ont des intérêts et des caractéristiques diversifiés. Deux groupes sont en interaction au sein de cette communauté : celui des producteurs et celui des utilisateurs.

Le modèle traditionnel du commerce basé sur le marché des biens matériels est le

³⁶<http://www.landesinteractives.net/> (rechercher dans le forum avec le mot clé logiciel libre).

fondement du logiciel propriétaire : le logiciel est considéré comme un bien économique matériel. Les relations entre les producteurs et les clients passent par l'intermédiaire des vendeurs qui sont au service des objectifs des propriétaires des moyens de production (les bureaux et les programmeurs qui y travaillent). Les éditeurs, propriétaires des moyens de production, déterminent les fonctionnalités du logiciel en fonction d'une planification de leurs revenus et des potentialités de vente perçues par les vendeurs : on appelle cela définir les besoins du marché. Les besoins peuvent le cas échéant être "orientés" par le marketing et les interdépendances voulues (pour inciter au renouvellement) ou subies (pour assurer une compatibilité) entre les versions et les différents logiciels complémentaires³⁷. Cette fonction intermédiaire de mise en marché consomme d'ailleurs bien plus de ressources que la fonction de production du logiciel (écriture et recopie) : une étude rapide des états financiers des éditeurs permet de constater que le coût de production est souvent nettement inférieur à 20% des revenus engendrés par le logiciel. Le "marché" est un médiateur important et déformant entre les utilisateurs et les programmeurs : ils ne sont pas en contact direct, il existe un monde intermédiaire, celui du commerce composé des vendeurs, des avocats, des conseillers de toute sorte (presse spécialisée, consultants ...) et des acheteurs. Aussi, le rôle des acheteurs est particulièrement important dans le cas des organisations qui sont les principaux clients de l'industrie du logiciel. Dans ce monde des (grandes) organisations publiques ou privées, les acheteurs ne sont pas les utilisateurs du logiciel : il existe ainsi une importante structure composée des vendeurs et acheteurs, souvent étanche et opaque, entre les programmeurs et les utilisateurs.

Dans le contexte du logiciel libre les deux communautés interagissent directement sans intermédiaire : c'est là probablement la principale différence entre le monde du logiciel propriétaire et celui du logiciel libre. L'évolution du logiciel est directement sous le contrôle des utilisateurs car il se diffuse et évolue dans la mesure où il satisfait à des besoins. Sans communauté d'utilisateurs prêts à contribuer d'une manière ou d'une autre à la production, aux tests et à la distribution, le phénomène du logiciel libre ne peut pas exister. Les deux communautés ne sont plus distinctes, elles s'intègrent l'une dans l'autre. C'est un phénomène difficile à comprendre pour quelqu'un profondément ancré dans la vision du logiciel basée sur celle des biens matériels classiques : il se voit comme un consommateur client n'ayant pas d'autre action sur le produit autre que sa décision d'achat. Dans le cas du logiciel libre l'utilisateur est un partenaire, plus ou moins direct, de la production du logiciel ou du moins de son évolution et de sa distribution. Les utilisateurs incapables de participer au développement jouent eux aussi un rôle important : celui de vendeur et de distributeur, en effet les utilisateurs satisfaits des logiciels libres en sont les principaux promoteurs et assurent souvent eux-même sa distribution à leur entourage. Ceci ne remet pas fondamentalement en question la notion de marché, mais en change l'interprétation : le monde du logiciel libre est un monde d'échange de travail, autrement dit de service. Nous avons affaire à une économie de troc ou de mise en commun du travail à tous les stades de la vie du logiciel : définition initiale, conception, réalisation, mise au point, promotion, distribution, assistance, évolution, toutes ces fonctions se distribuent "naturellement" au sein d'une

³⁷On utilise rarement un logiciel tout seul, mais un ensemble de logiciels qui doivent être compatibles entre eux.

communauté qui se crée et évolue avec le logiciel.

Dans la mesure où la fonction de vendeur disparaît, avec tout l'appareil du marketing et de la vente, le rôle des acheteurs dans l'organisation change fondamentalement : ils ne choisissent plus simplement un fournisseur (quand ils en ont la possibilité ! car le choix devant un monopole est forcément limité, on ne devrait alors plus parler de choix mais de justification ; seul reste un pouvoir de négociation sur le prix, quand cela est possible). Les acheteurs devraient participer à l'évolution et à la mise en oeuvre des logiciels, ils peuvent éventuellement rester en tant qu'intermédiaire mais en réalité devraient se fondre dans les deux communautés, celle des producteurs et celle des utilisateurs. Dans le cas du logiciel propriétaire la situation est claire, les intérêts économiques des uns et des autres sont bien définis, le vendeur doit dégager la plus grande marge bénéficiaire possible³⁸ et l'acheteur acheter le moins cher possible. Mais dans le cas des logiciels libres la donne est brouillée, le rôle des acheteurs est appelé à disparaître ou du moins à être transformé, en effet il reste des aspects juridiques et de négociations de contrats de service, ce qui est partiellement différent de la négociation de contrats de biens.

Le logiciel libre ne peut pas échapper aux contraintes économiques : les producteurs doivent recevoir d'une manière ou d'une autre un revenu, et ce revenu ne peut provenir que de la communauté des utilisateurs car les programmeurs producteurs sont soit des bénévoles (mais alors ils doivent avoir un autre revenu), soit plus généralement à l'emploi des organisations employant les utilisateurs de leurs logiciels, directement ou indirectement par le biais de la prestation de services. On n'assiste probablement pas là à une nouvelle économie basée sur la fraternité mais plutôt à une évolution du logiciel d'un mode de production-distribution basé sur le concept de bien matériel vers une organisation basée sur la notion d'économie de services (bien que le concept de fraternité n'en soit certainement pas absent). Il suffit de fréquenter les sites animés par les auteurs des logiciels libres pour avoir une idée de l'intégration des programmeurs et des utilisateurs qui caractérise le monde du logiciel libre.

3.5 Un phénomène en perpétuelle évolution

Le phénomène est complexe, mais il peut aisément s'interpréter quand l'on considère que les programmeurs sont eux aussi des utilisateurs de logiciel. L'intégration des deux groupes est facile à réaliser quand ils sont petits, homogènes et composés des même personnes, quand le programmeur est aussi l'utilisateur de ses propres logiciels. C'est en fait ce qui s'est passé au départ, le logiciel libre est né il y a plusieurs dizaines d'années dans de petites communautés de programmeurs.

Mais depuis, l'ampleur des logiciels a changé : il faut maintenant près de deux ans pour lire l'ensemble des instructions de Linux à raison de 24 heures par jour. L'utilisation des logiciels libres n'est plus limitée à la programmation. Certains utilisateurs mettent la main à la pâte, ce sont des enseignants, des scientifiques, des ingénieurs, des graphistes, ... qui possèdent l'expertise, parfois le talent, nécessaire à la production de logiciels de qualité. Les utilisateurs dits finaux peuvent participer à l'évolution du logiciel, en particulier à une opération coûteuse en temps : le débogage et la mise au

³⁸Son salaire en dépend d'ailleurs souvent très directement.

point.

Les utilisateurs ne sont plus uniquement des travailleurs au sein des organisations. De plus en plus ils ont un ordinateur à la maison, leur relation avec l'informatique a considérablement évolué au cours des dix dernières années, en grande partie grâce à l'essor d'Internet et de l'informatique graphique. Ceci a pour conséquence de brouiller encore un peu plus les cartes : ils s'attendent à utiliser les mêmes logiciels à la maison et au bureau (même s'ils ne les payent pas³⁹ !).

La relation entre d'une part les utilisateurs et d'autre part les producteurs se complexifiant, du fait du nombre de logiciels impliqués, et la taille des communautés grandissant, on a vu apparaître de nouveaux joueurs : les assembleurs et diffuseurs de distributions qui agissent dans un mode plus proche du service que des biens matériels avec des philosophies différentes, comme Red Hat⁴⁰, Suze⁴¹, Debian⁴², Mandrake⁴³, pour citer les plus connues et les plus utilisées (on peut en recenser plus d'une centaine⁴⁴). Ces distributeurs cherchent par différents moyens à intégrer les utilisateurs à l'évolution de leur produit, en général dans le cadre de la licence GPL. Ainsi la compagnie Mandrake a créé un club (payant)⁴⁵ de services et de rencontre des utilisateurs dans lequel, entre autres activités, elle organise la mise au point de paquetages au format RPM⁴⁶ afin de faciliter la mise à jour des logiciels : tous les membres du club peuvent voter pour les différents paquetages que certains d'entre eux ont proposé, puis des volontaires prennent en charge leur réalisation, d'autre (éventuellement ceux qui ont voté en faveur) les testent, par la suite ils sont disponibles pour tout le monde, membres du club ou non (ceux qui ont voté en faveur d'un paquetage sont avisés de sa disponibilité par courriel)⁴⁷. Ainsi l'intégration des producteurs et des utilisateurs se réalise sous différentes formes, ce peut être le soutien direct des gouvernements comme font certains pays d'Asie pour des groupes universitaires ou des groupes de recherche. En France, l'INRIA participe significativement à l'évolution de certains lo-

³⁹Récemment un de mes étudiants était étonné du fait que j'utilisais LyX pour générer mon syllabus en format pdf (depuis quelques temps je ne peux plus le déposer en format postscript sur le site web de ma faculté, seul le format pdf est accepté) et non Adobe Distiller, en discutant avec lui, je me suis aperçu il n'avait aucune idée du prix de ce logiciel et manifestement ne s'en préoccupait pas, il n'avait jamais envisagé que la licence de ce logiciel était payante !.

⁴⁰<http://redhat.com>

⁴¹<http://www.suse.com>

⁴²<http://www.debian.org/>

⁴³<http://www.mandrake.com>

⁴⁴Voir par exemple <http://www.distrowatch.com/>

⁴⁵<http://www.mandrakeclub.com/>

⁴⁶Un paquetage au format RPM est un fichier comprimé rassemblant tout ce qu'il faut pour installer automatiquement un logiciel. Ce format (libre) a été défini par la société Red Hat. Bien que créé par une société, Red Hat, il est utilisé par d'autres, Mandrake, et par de nombreux auteurs car le format et les logiciels d'installation sont distribués avec la licence GPL. Sur mon ordinateur la distribution Mandrake 9.0 a pu ainsi installer 3 versions du logiciel de gestion des paquetages compatibles entre eux car réalisés à partir des mêmes formats et codes : celui de Mandrake, celui de KDE et celui de Gnome (et il en existe d'autres !). Un aurait suffit, mais comme j'avais de l'espace disque et que j'étais curieux ... pourquoi me limiter ...

⁴⁷Actuellement le Club Mandrake met sur pied un serveur en accès limité à ses membres où ils peuvent facilement trouver tous les paquetages disponibles pour leur distribution. Ce serveur est réalisé avec des logiciels libres et contient des logiciels libres, ce qui est limité aux membres n'est pas le droit d'usage associé aux logiciels, mais la commodité de tous les trouver sur le même serveur.

giciels libres comme OpenLDAP et certains de ses chercheurs comme Bernard Lang⁴⁸ sont des ambassadeurs convaincus et actifs des logiciels libres.

Des logiciels propriétaires sont passés dans le monde libre comme Mozilla⁴⁹ (appartenant initialement à AOL, logiciel Netscape) ou OpenOffice⁵⁰ (appartenant initialement à Sun, logiciel StarOffice). De grands joueurs apparaissent, comme IBM qui, en tant qu'éditeur de logiciel, participe activement et significativement à l'évolution de Linux et inclut de plus en plus de logiciels libres dans son offre de logiciels. Les grands constructeurs comme IBM, HP ou Sun sont en train de construire une industrie du service basée sur les logiciels libres (IBM semble en avance sur les autres dans ce domaine), essentiellement autour de la construction des intranets et des extranets. Les mouvements entre logiciels libres et logiciels propriétaires sont de plus en plus nombreux et prennent de multiples formes, ce peut être la création d'une fondation pour la reprise d'un logiciel comme dans le cas de Blender⁵¹, ou la cohabitation d'un logiciel libre gratuit et d'une version payante (incluant des fonctions avancées ou du service) comme MySQL⁵² et "The MySQL AB Company" ou Argouml⁵³ dont la licence BSD autorise la commercialisation de Poseidon for Uml par Gentleware⁵⁴.

4 Comment se fait-il que les logiciels libres existent et évoluent ?

Lorsque l'on aborde la problématique des logiciels libres un ensemble de questions émerge généralement : comment peut-on pen

ser disposer gratuitement légalement d'un tel bien, quels sont les "communistes" à la base de cette folie (que l'on espère douce), comment peut-on espérer un travail cohérent d'une bande d'individus indépendants et totalement libres de faire ce qu'ils veulent, quel soutien et quelle pérennité peut-on espérer ? En effet l'existence non contestable des logiciels libres puissants et de qualité à la portée de tout le monde (ce texte a été totalement produit avec des logiciels libres⁵⁵) semble un paradoxe. Et cela l'est si l'on se contente de considérer le logiciel comme un bien commercial avec les yeux des membres du clan des personnes qui privilégient la satisfaction de leurs besoins à court terme et de leur confort. Mais le phénomène des logiciels libres devient parfaitement compréhensible, rationnel et souhaitable si on le considère d'un point de vue systémique (d'où notre analogie avec l'écologie) avec les yeux des membres du clan des personnes qui sont conscientes des dimensions collectives et des conséquences à moyen ou long terme de leurs choix et de leurs gestes.

⁴⁸<http://www.inria.fr/personnel/Bernard.Lang.fr.html>

⁴⁹<http://www.mozilla.org/>

⁵⁰<http://www.openoffice.org/>

⁵¹<http://www.blender3d.com/>

⁵²<http://www.mysql.com/>

⁵³<http://argouml.tigris.org/>

⁵⁴<http://www.gentleware.com/products/index.php3>

⁵⁵Il a été écrit avec LyX (<http://www.lyx.org>) exécuté avec Linux (distribution Mandrake 8.2, puis 9.0), et le furetage sur Internet était indifféremment fait avec Mozilla, Netscape ou Konqueror (le fureteur du bureau KDE).

4.1 Les conditions juridiques de la collaboration et de l'innovation

Le phénomène des logiciels libres repose fondamentalement sur le partage. Remarquons tout d'abord que les règles définies par les principes et les licences du logiciel libre existent pour garantir que personne ne puisse s'approprier indûment le travail des autres. Chacun doit pouvoir bénéficier du travail des autres mais nul ne doit pouvoir s'approprier le travail collectif. La FSF définit cette garantie du point de vue de la liberté, mais elle peut aussi s'interpréter comme étant une protection des intérêts de chacun dans un contexte de collaboration. Ne pas risquer de se faire piller son travail et avoir la garantie de pouvoir toujours bénéficier de l'ensemble du travail de la communauté sont en effet des conditions nécessaires à un travail collaboratif si chaque participant est indépendant. On comprend que dès que l'enjeu (la valeur d'utilisation du logiciel) est important pour chaque participant -ce qui justifie et garantit sa participation- il devient nécessaire de définir un cadre juridique approprié. C'est ce que propose la GPL et les licences de ce type.

On comprend alors pourquoi l'une des principales menaces qui pèse sur le phénomène des logiciels libres est la possibilité non plus de définir des copyrights portant sur le code lui-même, mais de déposer des brevets protégeant des procédés, c'est-à-dire des fonctions, et non plus une forme d'expression assurant la réalisation d'une fonction qui reste publique. Les communautés de développeurs sont riches en expertise mais collectivement pauvres en ressources monétaires, juridiques et administratives. Le dépôt d'un brevet est coûteux, en argent et en temps, et le risque est alors grand pour un individu expert de se voir voler son idée par une organisation riche et de devoir payer par la suite des droits pour utiliser son idée. Il faut bien comprendre que le préalable au dépôt d'un brevet est la non-existence préalable d'un brevet, et non l'existence préalable du procédé. Les brevets ont été définis dans le contexte de la production de biens matériels et non de biens immatériels liés au traitement de l'information, comme le furent le copyright qui protège l'expression d'une création intellectuelle.

On ne fait que commencer à comprendre le risque de dérive des brevets dans le domaine de la biologie, comme la possibilité de breveter un gène (et donc d'exiger des droits pour tout médicament ou traitement associé à ce gène) ce qui entraîne la possibilité de breveter une maladie, ou du moins son traitement ! C'est pour cela que certains soutiennent que la possibilité de déposer des brevets dans le monde du logiciel sera un frein à l'évolution et non une garantie de l'évolution. Les gouvernements prennent progressivement conscience de ce risque et certains ont pris, comme le gouvernement français, des positions claires allant à l'encontre de la brevetabilité des procédés informatiques⁵⁶.

4.2 Les conditions techniques et l'innovation technologique

Le développement des logiciels ou génie logiciel est un sujet complexe et plein de controverses, et le succès des logiciels libres ne le simplifie pas, en effet aux yeux de certains il invalide même certains dogmes relatifs aux bonnes pratiques (les fameuses

⁵⁶Un recensement de l'AFUL <http://www.aful.org/topic?subject=brevets>, Voir aussi l'interpellation de Bernard Lang http://www.lemonde.fr/imprimer_article_ref/0,9750,3244--268026,00.html

“best practices”). À ce sujet deux remarques s’imposent : tout d’abord des logiciels libres très complexes et volumineux innovatifs et de haute qualité existent (à commencer par Linux). Et deuxièmement, il faut bien remarquer que même dans le domaine des logiciels propriétaires ou privés (développés par une organisation à ses frais pour ses propres besoins) on est loin d’avoir fait l’unanimité sur la façon d’organiser et de gérer le développement des logiciels, il suffit pour s’en convaincre de constater le nombre de projets de logiciels qui se terminent mal, ou le nombre de méthodes disponibles ainsi que la difficulté que les analystes et les programmeurs ont à les suivre.

Les modes de fonctionnement des collectivités de développeurs de logiciels libres sont assez divers. Eric S. Raymond a fait une analyse très intéressantes de certains d’entre eux dans une série de trois articles très largement diffusés et commentés dans Internet et au cours de nombreuses rencontres⁵⁷. À la lumière de sa propre expérience de développeur du logiciel libre largement diffusé Fetchmail et en s’appuyant sur le développement de Linux et de la personnalité de Linus Torvalds, son créateur initial et animateur principal actuel, il montre comment un logiciel libre peut évoluer au sein d’une communauté de développeurs indépendants et volontaires mais qui adhèrent à un objectif commun. La métaphore de la cathédrale et du bazar qu’il utilise est très éclairante : le logiciel évolue progressivement en fonction des apports et des intérêts de chacun des contributeurs autour d’une idée commune et non en fonction d’un grand plan prédéterminé. Ce faisant il met en lumière certains principes qui se sont révélés redoutablement efficaces. Il est intéressant de remarquer que ces principes ne peuvent probablement pas être mis en oeuvre dans le contexte d’un développement propriétaire fermé de la même manière que dans un contexte de logiciels libres. Voici comment il résume son analyse dans son essai le “Le chaudron magique” :

1. Tout bon logiciel commence par gratter un développeur là où ça le démange.
2. Les bons programmeurs savent quoi écrire. Les grands programmeurs savent quoi ré-écrire (et réutiliser).
3. Prévoyez d’en jeter un, car de toute manière, vous le ferez. (Fred Brooks, “The Mythical Man-Month”, chapitre 11)
4. Si vous avez la bonne attitude, les problèmes intéressants viendront à vous.
5. Quand un programme ne vous intéresse plus, votre dernier devoir à son égard est de le confier à un successeur compétent.
6. Traitez vos utilisateurs en tant que co-développeurs est le chemin le moins semé d’embûches vers une amélioration rapide du code et un débogage efficace.
7. Distribuez tôt. Mettez à jour souvent. Et soyez à l’écoute de vos clients.
8. Étant donné un ensemble de bêta-testeurs et de co-développeurs suffisamment grand, chaque problème sera rapidement isolé, et sa solution semblera évidente à quelqu’un.
9. Il vaut mieux avoir des structures de données intelligentes et un code stupide que le contraire.

⁵⁷La cathédrale et le bazar : http://www.linux-france.org/article/these/cathedrale-bazar/cathedrale-bazar_monoblock.html, À la conquête de la noosphère http://www.linux-france.org/article/these/noosphere/homesteading-fr_monoblock.html, Le chaudron magique : http://www.linux-france.org/article/these/magic-cauldron/magic-cauldron-fr_monoblock.html.

10. Si vous traitez vos bêta-testeurs comme ce que vous avez de plus cher au monde, ils réagiront en devenant effectivement ce que vous avez de plus cher au monde.
11. Il est presque aussi important de savoir reconnaître les bonnes idées de vos utilisateurs que d'avoir de bonnes idées vous-même. C'est même préférable, parfois.
12. Bien souvent, les solutions les plus innovantes, les plus frappantes, apparaissent lorsque vous réalisez que votre approche du problème était mauvaise.
13. La perfection est atteinte non quand il ne reste rien à ajouter, mais quand il ne reste rien à enlever.
14. Tout outil doit être utile par rapport aux utilisations qu'il a été prévu d'en faire. Mais on reconnaît un outil vraiment excellent au fait qu'il se prête à des usages totalement insoupçonnés.
15. Quand vous écrivez un logiciel jouant le rôle d'une passerelle quelconque, prenez soin de perturber le moins possible le flot de données – et ne perdez jamais d'éléments d'information, à moins que la machine destinataire vous y oblige !
16. Quand votre langage est loin d'être Turing équivalent, un peu de "sucre syntaxique" ne peut qu'aider.
17. Un système de sécurité n'est pas plus sûr que le secret (la clé) qui le garde. Gare aux pseudo secrets !
18. Pour résoudre un problème intéressant, commencez par trouver un problème qui vous intéresse.
19. Pour peu que le coordinateur du développement dispose d'un moyen de communication au moins aussi bon que l'Internet, et pour peu qu'il sache comment mener ses troupes sans coercition, il est inévitable qu'il y ait plus de choses dans plusieurs têtes que dans une seule.

Une bonne partie de ces principes ont été appliqués dans la plupart des logiciels libres à succès de manière étonnamment différente. Il n'y a donc pas qu'une recette magique, mais bien plusieurs et elles n'ont certainement pas toutes été inventées. Cependant, tous les logiciels⁵⁸ ne se prêtent pas forcément à un développement libre, il faut qu'une communauté d'utilisateurs existe ou puisse se développer, mais aussi que le logiciel, ou le besoin ou problème, réponde a priori à certains critères.. C'est ainsi que Eric S. Raymond toujours dans son essai "Le chaudron magique" dégage les critères suivants, plus le logiciel satisfait à ces critères plus les chances de lever seront grandes :

1. lorsque la fiabilité/stabilité/pérenité du logiciel sont critiques
2. lorsque la qualité de la conception et de l'implémentation du logiciel sont difficilement vérifiables par un autre moyen que l'analyse critique de ses pairs
3. lorsque le logiciel est indispensable à l'activité du client
4. lorsque le logiciel établit ou met en oeuvre une infrastructure banale, d'informatique ou de communications

⁵⁸Ce n'est pas l'avis de Richard M. Stallman pour qui tous les logiciels devraient être libres, pour lui le concept de logiciel propriétaire est fondamentalement mauvais.

5. lorsque les algorithmes clefs du logiciel (ou leurs équivalents fonctionnels) font partie de la base de connaissances communes des informaticiens

Ainsi il apparaît de plus en plus clairement que si le succès du développement et de l'évolution d'un logiciel libre ne peut pas être garanti en terme technique avec certitude, la situation n'est pas pire que celle qui prévaut dans le monde des logiciels propriétaires, au contraire, elle serait même favorable.

En ce qui concerne l'innovation, force est de reconnaître que les logiciels libres sont à l'origine de bien des innovations, en particulier celles qui concernent Internet, et que les arguments tendant à démontrer que les logiciels libres découragent l'innovation relèvent plus de la propagande que de la réalité et de la confusion entre les fonctions d'éditeur et de créateur ou de programmeur.

4.3 Le rôle des standards et normes : une vision historique de l'évolution vers le service

On peut considérer que le développement de l'informatique dans les organisations s'est déroulé jusqu'à maintenant en trois phases, chacune étant dominée par un secteur d'activité industrielle : la première phase a été dominée par les fabricants de matériel, la deuxième par les éditeurs de logiciels, et la troisième, qui est en émergence, est celle des services. Cela se voit très bien en considérant l'évolution de la structure des budgets que les organisations consacrent aux systèmes informatisés. La part du matériel a décliné progressivement en passant du poste le plus important au poste le moins important. La part du logiciel, prépondérante au cours de la deuxième phase, décroît actuellement au profit de la part des services (installation, assistance, formation, ajustement, maintenance, ...). Un examen du TCO⁵⁹ est très instructif à cet égard, on y constate de plus en plus fréquemment que la mise en oeuvre d'un réseau de postes de travail de bureau ou l'implantation d'un package coûte nettement plus cher que l'acquisition des licences des logiciels ou que le matériel.

La constatation de cette évolution a surpris bien du monde ; qui aurait prédit la disparition de manufacturiers comme Univac ou DEC au moment de leur apogée ? Il risque d'en être de même pour les éditeurs de logiciels qui ne sauront pas prendre le virage du service. On comprend ainsi l'attitude d'IBM, et plus récemment celle de Oracle ou de Sun vis-à-vis des logiciels libres (notamment Linux) ainsi que celle bien différente de Microsoft. Il suffit, pour se convaincre de la réalité de cette évolution de considérer le comportement d'un joueur majeur comme IBM pour avoir la confirmation de cette évolution. Si IBM a probablement subi contre son grès (à cause de l'application de la loi anti-trust des États-Unis) le passage de la première à la deuxième phase, elle a manifestement anticipé le passage à la suivante et se trouve actuellement en position de réussir le passage de la deuxième à la troisième phase comme le montre sa stratégie vis-à-vis Linux et les logiciels libres ainsi que l'évolution de son activité (la part des revenus provenant des services devient la part principale)⁶⁰.

⁵⁹Total Cost of Ownership : coût total de possession. Ce coût inclut le matériel, le logiciel, mais aussi son installation, sa maintenance (redevance et personnel propre) et souvent la formation. Voir par exemple : Why Open Source Software / Free Software (OSS/FS) ? Look at the Numbers ! de David A. Wheeler http://www.dwheeler.com/oss_fs_why.html

⁶⁰L'engagement de IBM en ce qui concerne les logiciels libres, la consultation et les services de support est

Un facteur déterminant du passage d'une phase à l'autre est celui des standards et normes. Le passage de l'ère de domination du matériel à celui du logiciel n'a pu se faire que grâce à l'apparition de la normalisation des langages de programmation (comme COBOL) qui rendaient (dans une certaine mesure, bien entendu) la programmation indépendante de la plate-forme matérielle (l'amélioration considérable du rapport coût/performance y est bien sûr aussi pour quelque chose). Pour qu'une norme joue son rôle il faut qu'elle soit stable, acceptée et sous le contrôle de l'ensemble de la communauté concernée⁶¹. Le passage de l'ère du logiciel à celle du service repose lui aussi sur l'existence de standards et normes, notamment au niveau des échanges : Java, Posix, C++, TCP/IP, XML⁶² par exemple sont fondamentaux. C'est pour cela qu'il est si important de ne pas les laisser ainsi que des organisations comme W3C "tomber sous la coupe" d'un fournisseur dominant quel qu'il soit.

Le respect de tels standards et normes garantit la pérennité de la mise en oeuvre des logiciels, autrement dit l'indépendance des fonctions par rapport au logiciel. Il faut mettre ceci en parallèle avec l'indépendance du logiciel par rapport aux plates-formes matérielles. Cette standardisation publique n'est évidemment pas à l'avantage des éditeurs de logiciel qui ont tout intérêt à empêcher le remplacement de leurs logiciels par un logiciel concurrent qu'il soit libre ou propriétaire. Également, dans la recherche de la croissance de leur chiffre d'affaire, les éditeurs de logiciels ont tout intérêt à forcer le remplacement des versions par de nouvelles, même si cela impose un coût de mise en oeuvre supplémentaire à leurs clients sans un véritable gain sur le plan des fonctionnalités. On comprend ainsi la "bataille pour le contrôle des normes et standards" par les joueurs dominants et l'exaspération des responsables informatiques des grandes entreprises qui voient une partie importante de leurs ressources (argent et temps) consacrées à la mise à jour forcée de versions sans augmentation de service auprès de leurs utilisateurs.⁶³

C'est ainsi que l'on peut parler de "non décision" en ce qui concerne l'acquisition de certains logiciels. On peut le vérifier facilement dans bien des organisations en demandant qui a décidé du passage de la version X à la version Y du logiciel W associé au changement de la version 1 à la version 2 du logiciel Z⁶⁴, ce qui entraîne

clairement présentée dans son rapport annuel de 2000 http://www.ibm.com/annualreport/2000/flat/toc/2_3_1_intro.html

⁶¹D'après le Dictionnaire encyclopédique de l'information et de la documentation, éditions Nathan, collection "Réf", Paris, 1997, la définition d'une norme est la suivante : « La norme est un document établi par consensus et approuvé par un organisme reconnu, qui fournit, pour des usages communs et répétés, des règles, des lignes directrices ou des caractéristiques, pour des activités ou leurs résultats, garantissant un niveau d'ordre optimal dans un contexte donné ».

⁶²Pour une introduction à XML, voir le chapitre 2, introduction à XML et XSLT, de l'ouvrage "Comprendre XSLT Comprendre XSLT de Bernard Amann et Philippe Rigaux, O'Reilly, accessible à l'adresse http://www.oreilly.fr/catalogue/comprendre_xslt.html

⁶³Voir par exemple les mouvements d'humeur de Jean Paul Corniou, président du CIGREF (Club Informatique des Grandes Entreprises Françaises) : <http://www.cigref.fr>, <http://www.linuxfr.org/2002/04/06/7829,0,1,0,0.html> et aussi http://solutions.journaldunet.com/0109/010928_cigref.shtml : "Au sujet du business model des grands éditeurs et constructeurs informatiques, Jean-Pierre Corniou n'a pas hésité à qualifier d'aberrante la situation dans laquelle ces derniers plaçaient trop souvent leurs clients. Il évoque un "modèle d'obsolescence planifiée"... "Habités à des taux de croissance de 30% par an et à des marges de l'ordre de 30 à 50%, les vendeurs informatiques n'achèvent pas ce qu'ils commencent...", a déploré le président du Cigref"

⁶⁴Nous laissons à l'imagination du lecteur le soin de mettre des noms afin d'illustrer ceci

en outre un changement de matériel (à cause de la demande de puissance et de mémoire supplémentaire), sur la base de quelle étude et quels sont les bénéfices ? Bien (trop ?) souvent on ne reçoit pas de réponse à ces questions. C'est pour cela que l'on peut parler de "non-décision", la décision a été imposée par l'environnement (principalement les fournisseurs de logiciels grâce à la maîtrise des licences et des standards ou normes). Par exemple l'achat de nouvelles licences ne peut se faire que pour les nouvelles versions, forçant ainsi à mettre à jour pour des raisons de compatibilité (réelle ou simplement revendiquée) les anciens postes, ou encore la version 1 du logiciel Z qui n'est plus entretenue par son fournisseur que sur la version A du logiciel W, etc.

Ainsi, grâce à l'inter-opérabilité procurée par les normes ou standards, le logiciel devient une commodité au même titre que le matériel, et le nerf de la guerre (qui consomme l'essentiel des ressources) devient le service. Mieux encore, ceci commande de considérer le logiciel comme un service et non comme un produit, or il se trouve que la pensée commerciale "produit" est difficilement compatible avec la pensée commerciale "service" : au delà des caractéristiques des produits et de la satisfaction du client, l'organisation de la vente d'un logiciel propriétaire repose sur le marketing et la structure des commissions de vente alors que l'organisation d'une activité de service repose sur l'expertise et la renommée des agents de service. Promouvoir et vendre un service n'est pas identique à promouvoir et vendre un produit : en général on ne s'intéresse pas aux CV des concepteurs et des programmeurs des logiciels propriétaires dominants alors que l'argument de vente principal du service repose sur eux.

4.4 Les conditions socio-économiques

Pour se convaincre que les conditions économique du développement et de la diffusion des logiciels libres n'a pas grand-chose à voir avec celles des logiciels propriétaires, il suffit de considérer ce simple fait : certaines communautés -et non des moindres⁶⁵- se financent (en tout ou partie) par la vente de casquettes, T-shirts ou de tasses ! On parle bien de vente et non de distribution gratuite.

L'économie de la production du logiciel est différente de celle de la production des biens physiques. Un simple examen comptable de la situation suffit pour le comprendre. Le coût de reproduction une fois le produit réalisé est très faible, sinon nul, pour le fournisseur (c'est de plus en plus le client qui réalise lui-même les copies des versions qu'il utilise). Ceci pose un problème particulier pour la fixation des prix de vente et peut expliquer la singularité, pour ne pas dire la dérive, observée dans l'ensemble de l'industrie du logiciel propriétaire : les coûts de production sont proportionnellement faibles en regard des coûts de vente ou des revenus des propriétaires, si on compare cette industrie aux autres industries légales on a une structure de coûts qui s'apparente aux produits de luxe⁶⁶. On peut ainsi calculer à partir de la lecture des rapports annuels des éditeurs de logiciel la proportion du prix payé consacrée au développement des logiciels qu'on achète. L'exercice est instructif : prenons le cas d'un logiciel de type ERP produit et distribué par une entreprise dans laquelle 20 % du personnel est affecté au développement et aux tests ; ceci peut représenter, disons, 10

⁶⁵<http://www.gnu.org/order/t-shirts.html>

⁶⁶Des esprits chagrins ont parfois comparé la structure des coûts et revenus de certaines entreprises de logiciel à celles du commerce de la drogue.

% des revenus (les propriétaires, cadres et les vendeurs sont en général mieux payés que les programmeurs ou le personnel de l'assurance qualité) ; on n'utilise en général qu'une partie des fonctions acquises, disons 50 %, ces fonctions incluent pour des raisons d'indépendance d'autres fonctions normalement assurées par le SGBD (qu'il faut de toute façon acquérir) ; pour des raisons d'évolution et de complexité le développement du logiciel acquis est plus complexe (donc plus long) qu'un développement sur mesure ; finalement on peut estimer que 100 \$ de logiciel acquis correspond au plus à 1 \$ de développement utile. Bien sûr ce calcul soit un peu caricatural, il illustre bien une certaine réalité et il met en lumière que ce que le client achète est de la vente plus que du logiciel. Par exemple le développement du chiffrier Lotus a coûté 300 000 \$ alors que la première campagne de publicité a coûté 4 200 000 \$⁶⁷. Ce qui ressort de ceci est que ce n'est pas la création du logiciel qui coûte le plus cher.

Certains pensent, ou clament que l'essor du logiciel libre risque de détruire l'économie. Mais de quelle économie parle-t-on ? De celle d'un pays ? de celle des programmeurs de logiciel ? de celle du logiciel propriétaire ? Il est vrai que l'économie des pays occidentaux est de plus en plus basée sur l'information ; les économistes calculent que plus de la moitié l'activité économique des pays développés touche de près ou de loin à l'information. C'est vrai, mais assimiler cette économie à celle du logiciel propriétaire est un amalgame inacceptable. La fabrication du logiciel ne concerne qu'une petite partie de l'économie de l'information : il y a aussi l'enseignement, les loisirs et l'art, l'administration des organisations et des organismes publics de tous les niveaux du gouvernement, etc. Toutes ces activités ne se limitent pas à de la production de logiciel ! Et de plus, quand elle repose sur du logiciel, elle n'a rien à voir avec l'origine des logiciels qu'elle utilise. Qui plus est le logiciel propriétaire ne concerne qu'une minorité des programmeurs et des analystes, Eric S. Raymond l'estime à moins de 5% des programmeurs. Il est vrai cependant que les programmeurs ne forment pas la classe d'employés la plus nombreuse des éditeurs de logiciels. Il est impossible de démontrer une quelconque réduction du nombre des personnes employées à la production et à l'entretien des logiciels dans l'ensemble des entreprises publiques et privées même si les logiciels libres étaient largement prédominants. L'employeur de ces programmeurs peut changer, mais ils continueront à avoir un emploi. Les emplois peuvent se déplacer, d'une entreprise à l'autre ou même d'un pays à l'autre, dans la mesure où les emplois créés et entretenus dans le contexte des logiciels libres sont directement liés à l'utilisation du logiciel ; les emplois des entreprises de services de dispersés dans le territoire desservi et non plus concentrés dans certains lieux de production.

Le succès des logiciels libres entraîne un certain changement dans la structure de l'économie du logiciel mais en aucun cas sa disparition. C'est bien ce qu'ont compris certains pays asiatiques et européens : le logiciel libre est pour eux un moyen de contrer les effets néfastes de l'hégémonie de certains grandes corporations de plus en plus en situation de monopole. Ils ont compris qu'un pays s'enrichit en procurant des emplois technologiques à ses citoyens mais qu'il s'appauvrit quand il paye des redevances à une firme étrangère, mais qu'à l'inverse un pays peut continuer à s'enrichir s'il réussit à contraindre les autres pays à payer une redevance pour l'usage de ses logiciels. L'équation est simple à établir et à comprendre. Il est intéressant à ce sujet de remar-

⁶⁷Comme j'ai pu le lire dans un article de Byte, qui a disparu de ma bibliothèque ...

quer que le phénomène des logiciels libres ignore les frontières nationales, ceci peut être un problème pour certains pays qui veulent créer une “industrie” nationale exportatrice du logiciel. Il est certain qu’advenant une large diffusion des logiciels libres, le secteur économique le plus (et probablement le seul) vraiment touché (en terme de réduction du volume d’activité) est celui de la promotion et de la vente des logiciels. Les secteurs liés à cette activité de vente comme les magazines spécialisés, les firmes de conseil technologiques, les foires commerciales seraient eux aussi touchés, mais probablement plus en terme de *modus operandi* (plus de formation et d’information, moins de publicité et de communiqués) que de volume d’activité.

Les études actuelles montrent qu’une proportion de plus en plus importante des programmeurs contribuant aux logiciels libres sont payés par leur entreprises pour participer à leur développement⁶⁸. La firme IBM, a annoncé qu’elle allait consacrer un milliard de dollars à Linux et aux outils de développement des applications de gestion basés sur Linux. Ainsi on peut constater que d’ores et déjà plusieurs entreprises, soit en tant que grosses utilisatrices de logiciels, soit en tant que joueur majeur sur le marché des logiciels libres (IBM, HP, Sun, AOL-Netscape, ...) participent significativement à l’essor des logiciels libres. Si IBM, firme championne du monde pour le dépôt de brevets (la plupart concernant les composantes et le matériel), participe à ce mouvement, ce n’est probablement pas par philanthropie, mais parce qu’elle perçoit l’évolution du marché vers le service.

Mais le logiciel libre ne concerne pas que les systèmes d’exploitation et les logiciels d’application généraux à grande diffusion comme les outils de développement (PHP par exemple) ou les suites bureautiques. Il concerne aussi les logiciels d’application, c’est ce qui commence, par exemple, à être sérieusement envisagé par les institutions publiques européennes⁶⁹. Les deux exemples (actuellement hypothétiques mais très réalistes) développés ci-dessous illustrent bien comment un développement libre peut être une solution de bonne gestion.

Premier exemple concret

Les universités utilisent de plus en plus des logiciels de “e-learning” (apprentissage en ligne); ce sont des intranets supportant l’activité pédagogique, ils permettent de gérer et supporter l’activité des classes d’étudiants en relation avec les professeurs.

Il existe des logiciels propriétaires pour ce faire⁷⁰ avec les avantages habituels perçus (fonctionnalités supérieures à celles qu’une université peut développer elle-même, garantie apparente d’un service et de l’évolution des versions...) et les inconvénients habituels (difficulté de personnalisation et d’intégration aux systèmes existants -aggravée par le caractère fermé du logiciel-, cadence imposée des mises à jour, évolution peu contrôlable des coûts ...).

⁶⁸Plus du tiers des programmeurs participant au logiciels libres ayant répondu à l’enquête conduite par l’Université de Maastricht sont payés par leurs employeurs pour ce faire. <http://www.infonomics.nl/FLOSS/report/Final0.htm>

⁶⁹Pooling open source software <http://europa.eu.int/ISPO/ida/jsps/index.jsp?fuseAction=showDocument&parent=news&documentID=550>

⁷⁰Parmi les nombreux logiciels existants, certains commencent à avoir une position dominante, au point que l’on peut se demander si l’on assiste pas ici encore une fois à l’évolution vers un monopole.

Supposons (de façon réaliste réaliste) que le développement d'un tel produit nécessite un effort d'une quinzaine d'années personne pour la première version et d'une dizaine d'années personnes pour chaque version (sur une base annuelle). Or chaque université consacre des ressources à la mise en oeuvre et à la personnalisation de l'intranet, disons une demi-douzaine de personnes (c'est largement sous-estimé).

Si quelques universités (disons une demi-douzaine) s'entendaient pour développer un logiciel libre (en licence GPL) en y consacrant la moitié de leurs ressources consacrées aux intranets (plus celles dégagées par les économies de licence) en partant d'un logiciel libre existant (il en existe actuellement plus d'une douzaine sans compter les intranets non spécialisés pour l'enseignement mais ayant des fonctions réutilisables comme les logiciels de forum), il devient évident qu'elles bénéficieraient d'un meilleur produit, mieux adapté, plus évolutif et qu'elles auraient entre elles (par nécessité) la garantie de la pérennité du logiciel. On ne parle pas ici de bénévolat, ni de communisme, mais de bonne gestion.

Ceci suppose bien sûr que les universités ne se considèrent pas en concurrence au point de penser tirer des avantages compétitifs significatifs grâce à leurs environnement de "e-learning". Ceci suppose qu'elles s'entendent, en fait cette entente peut émerger naturellement grâce aux liens qui s'établissent entre développeurs grâce à Internet, c'est le pendant de la non-décision évoquée ci-dessus. Ce scénario est aujourd'hui loin d'être irréaliste, s'il n'est pas plus pratiqué dans toutes les universités c'est probablement pour des raisons d'ignorance du phénomène des logiciels libres.

Deuxième exemple

Il concerne l'équipement en logiciel des cabinets de médecin. Dans un système de santé comme celui du Québec, si l'on voulait assurer une bonne circulation de l'information dans un objectif de continuité des soins entre les cabinets de médecins et les diverses institutions de production de service de santé, il faudrait que tous (ou presque) les cabinets privés soient au moins partiellement informatisés pour assurer la collecte, l'émission, la réception et l'utilisation des informations communes assurant un minimum de qualité en ce qui concerne la continuité des soins. Comment assurer cette informatisation, quand l'on sait qu'actuellement une faible proportion des cabinets sont informatisés (disons 10%) et qu'il n'y a peu de concertation entre les développeurs des divers logiciels installés dans ces cabinets et ceux installés chez les différents prestataires de service ?

La première étape consiste à définir à partir d'un modèle de données public commun, la structure des échanges de données. Supposons que ceci soit fait, il faut maintenant développer les logiciels. Faisons une première hypothèse : le recours à des logiciels commerciaux. Dans ce cas, pour assurer la compatibilité, il faudrait que le ministère responsable mette en place un bureau d'accréditation. C'est un travail délicat qui risque de coûter plusieurs années personne, pour la mise en place, puis le faire fonctionner (pensons aux modifications des échanges qui risquent de survenir et d'alourdir la procédure : qui assure les coûts -et les honoraires d'avocat- de modification des systèmes déjà installés ?). Les entreprises vont alors se lancer dans le développement. Supposons qu'un tel développement pour obtenir un produit commercial, correctement documenté, nécessite cinq années personnes (c'est très raisonnable comme estimation).

Ceci va représenter un investissement de l'ordre de 1 000 000 \$ (j'arrondis), il faudrait alors que le produit génère des ventes de 5 000 000 \$ pour qu'il soit rentable (pensons aux frais de vente, aux licences incorporées, à l'accréditation, aux tests de compatibilité, ...). Supposons qu'il faille équiper 10 000 cabinets (j'arrondis encore un peu, mais pas trop). Admettons qu'il y ait quatre produits concurrents (on voudrait probablement éviter de se trouver en situation monopolistique), il faudrait vendre chaque logiciel au moins 2 500 \$ plus l'entretien et les frais d'installation, c'est trop cher, diront alors les médecins ! De plus, cela prendrait probablement de 3 à 5 ans : on peut penser que c'est trop long.

Considérons une autre stratégie, celle des logiciels libres. On démarre un projet libre sous la supervision du ministère responsable de façon à s'assurer du respect des structures de données et à amorcer le financement. Comme on lance un projet libre, on ne part pas de rien, il existe déjà des projets libres, autant s'en inspirer⁷¹. Le ministère responsable peut financer deux à trois années personnes (de toute façon il devait le faire pour mettre en place l'accréditation) et s'assurer la coopération de quelques médecins (j'en connais prêts à le faire). Ces derniers sont indispensables, on ne peut pas réaliser un tel projet sans les impliquer (ce sont les premiers utilisateurs). Associons à ce projet quelques chercheurs dans les universités (qui ont des étudiants à la recherche de projets stimulants). Il y a fort à parier que l'on aura au bout d'un an un très bon logiciel libre, gratuit, prêt à être déployé non pas par les fonctionnaires du ministère responsable mais par des sociétés de service qui factureront leur travail d'assistance à l'installation et à l'entretien du logiciel (et éventuellement amélioreront le logiciel commun mais en licence libre, disons GPL).

Ces deux exemples sont-ils des rêves d'extrémistes communistes ou de saine gestion ? On ne parle pas de bénévolat mais de performance technique et économique en regard d'un objectif de service : c'est cela de la bonne gestion.

5 Pour une évolution vers l'usage des logiciels libres

5.1 Le logiciel est un bien immatériel

Le logiciel est un bien immatériel typique de la société de l'information qui répond à des caractéristiques différentes du monde matériel reposant sur la rareté des matières premières et de l'énergie :

- c'est un bien non rival, son utilisation ne le détruit pas, on peut même dire que parfois plus il est utilisé, plus il a de la valeur ;
- son coût marginal est nul, en produire un autre exemplaire ne coûte que la copie (uniquement le coût de la communication dans la plupart des cas, qui est d'ailleurs de plus en plus à la charge de l'utilisateur).
- il requiert un coût d'investissement faible (matière grise) en regard des grands investissements matériels (comme une route ou une nouvelle automobile).
- il a un effet réseau, sa valeur dépend du nombre des utilisateurs, plus il y a d'utilisateurs, plus il est attrayant pour un nouvel utilisateur.

⁷¹Par exemple suivre les liens de ce recensement de projets médicaux libres : http://www.linuxfrench.net/article.php3?id_article=1032

- il a un effet d’inertie, plus on l’a utilisé, plus on est lié au produit car en changer implique un coût élevé de transition.

Ces caractéristiques font que pour une fonction donnée, au début plusieurs logiciels sont en concurrence car le droit d’entrée est alors faible. Puis, progressivement quelques un vont s’imposer et finalement à cause des effet réseau et d’inertie, l’un d’entre eux (pas forcément pour des raisons de qualité technique, ni de prix) devient dominant. Le droit d’entrée devient alors énorme et aucun concurrent ne peut plus envisager de pénétrer ce marché. Ces caractéristiques conduisent quasi fatalement à une situation monopolistique, avec toutes les conséquences liées aux monopoles, qui s’accompagne en particulier comme on le démontre de plus en plus d’une baisse du rythme de l’innovation en raison des barrières techniques destinées à protéger les intérêts des propriétaires.

Curieusement ces mêmes conditions⁷² ont entraîné la naissance de l’antidote à cette aboutissement monopolistique : le logiciel libre. Basé sur des normes ouvertes et publiques, le logiciel libre permet l’émergence de nouvelles solutions et de l’innovation, il redonne aux petits la possibilité de se développer sans risque d’être avalé par l’entreprise monopolistique dominante.

5.2 Un logiciel repose sur des normes

Un logiciel repose sur deux ensembles de normes : celles qui définissent l’accès à ses fonctions et celles qui définissent les documents (qui contiennent des informations sous de multiples formes) qu’il lit ou écrit. Une norme publique est le meilleur moyen de limiter les coûts de transition qui accompagnent le changement ou même l’évolution des logiciels.

Actuellement la norme XML est en voie d’être adoptée par de plus en plus de gouvernements et d’entreprises. Dans ce domaine les logiciels libres sont en avance et sont la meilleure garantie du respect du caractère public et de la pérennité des documents créés en suivant cette norme. C’est ainsi que les logiciels libres de bureautique disponibles ou en développement privilégient ce format (OpenOffice, Kword, ...).

Linux repose (même si ce n’est pas officialisé) sur la norme Posix, seule norme publique en ce qui concerne les systèmes d’exploitation, c’est la meilleure garantie dont l’on puisse actuellement disposer sur la stabilité des fonctions d’un système d’exploitation. Cette garantie est importante car tout changement de système d’exploitation est susceptible d’entraîner de grands coûts d’évolution ainsi que peuvent actuellement le constater les entreprises qui migrent de W98 à W2000 ou W/XP.

5.3 Un logiciel libre n’est pas (simplement) un logiciel gratuit

L’absence de coût de licence, bien qu’elle soit importante (elle a motivé et motive encore bien des gens à opter pour des logiciels libres) n’est pas la seule dimension qui justifie le recours aux logiciels libres.

⁷²il est intéressant de remarquer que les sociétés (primitives) qui fonctionnaient d’une manière analogue au logiciel libre impliquaient une faible population en regard des ressources disponibles ce qui rendait par exemple la nourriture un bien non rival à condition de suivre quelques règles communes qui n’ont rien à voir avec le droit de propriété individuelle.

- La gratuité de l'accès au logiciel libre n'est pas sa caractéristique essentielle : ce qui fait qu'un logiciel est libre c'est l'accès au code source, le droit d'en faire ce qu'on veut sans rien devoir à personne, mais aussi sans pouvoir s'appropriier le logiciel.
- Le logiciel libre repose par définition sur des normes ou standards libres et ouverts, mais pas obligatoirement sur des normes (et actuellement en général sans accréditation officielle à cause des coûts liés à cette accréditation ; mais rien n'interdit un gros utilisateur de réaliser cette accréditation).
- Un logiciel libre repose sur une communauté dynamique de producteurs et d'utilisateurs. Sans cette communauté le logiciel est mort, il n'évolue plus. Par contre une communauté active est la meilleure garantie de la pérennité d'un logiciel.
- Le contexte juridique du logiciel libre définit et garantit les conditions et la pérennité de coopération. Face à la complexité engendrée par les diverses licences du monde des logiciels libres, il apparaît de plus en plus que les administrations devraient actuellement privilégier la licence GPL ; c'est celle qui garantit le mieux la pérennité du logiciel et diminue les risques juridiques. Il faut s'attendre à ce que la défense des éditeurs de logiciels propriétaires face aux logiciels libres va se faire de plus en plus sur le terrain juridique.

5.4 Le logiciel (qu'il soit libre ou propriétaire) n'est pas simplement un bien commercial que l'on acquiert

Le prix d'acquisition d'un logiciel ne représente somme toute qu'une petite partie des coûts reliés à la manipulation (création, communication, conservation) des informations (données et documents) à l'aide de l'informatique dans la vie d'une organisation. Au plan commercial il est tout de même étonnant que de plus en plus d'observateurs trouvent normal de comparer les produits de Microsoft, une riche compagnie⁷³ d'environ 40 000 employés à celui distribué par une petite compagnie de moins d'une centaine d'employés comme Mandrake⁷⁴. Cette comparaison est pour le moins boiteuse. En effet, une telle distribution qui s'installe en général en moins d'une heure et nécessite un seul redémarrage inclut non seulement Linux mais aussi un serveur, un pare feu, plusieurs suites bureautique, des logiciels graphiques, des outils de développement, une ou plusieurs base de données relationnelle, plusieurs centaines d'utilitaires (de toutes sortes, incluant des convertisseurs de document dans la plupart des formats existants), des jeux ... soit plus d'un giga octet de logiciels (ceci représente plusieurs dizaines de milliers de dollars en terme de logiciels propriétaires) parfaitement fonctionnels et installés en quelques minutes ! D'autre part, les logiciels qui composent une distribution assemblée par les employés du groupe la réalisant proviennent d'une large communauté de développeurs et de testeurs⁷⁵⁷⁶.

⁷³CA Microsoft : 26 Milliards de US \$ et 10 Milliards de US \$ de marge nette en 2001

⁷⁴3,5 millions euros de chiffre d'affaires et -13,5 millions de résultat en 2000/2001 <http://www.mandrakesoft.com/company/investors/financials>

⁷⁵En contre partie les logiciels développés par les employés de Mandrake (utilitaires d'installation) sont retournés à la communauté car ils sont couverts par la licence GPL et librement accessibles sur le site Internet de la compagnie.

⁷⁶Si vous voulez en avoir rapidement une illustration convaincante des logiciels libres, essayez donc Knoppix (<http://knoppixfr.tuxfamily.org/>) une distribution Linux sur CD, vous n'avez même pas besoin de

Ainsi :

- Le logiciel n'est pas une fin en soi, c'est un moyen pour créer, communiquer, conserver, manipuler ... des informations de toutes sortes, que ce soient des données formatées dans des documents structurés (des formulaires aux bases de données), ou des documents de nature plus ou moins complexes incluant des composantes de diverses natures (textes, dessins, photos, ...). Dans cette perspective, les documents manipulés sont plus importants que les logiciels, ceci conduit à comprendre que les formats des documents importent plus que les logiciels : les standards et normes sont fondamentaux et l'on comprend de plus en plus pourquoi ils devraient être "ouverts", c'est à dire publics et accessibles à tous sans redevance à leurs auteurs, ce qui est une des conditions de la garantie de leur pérennité. Ainsi le critère numéro un du choix d'un logiciel devrait être le standard ou la norme définissant les documents qu'il manipule (importation, stockage et exportation).
- Un logiciel est (presque) toujours exploité et entretenu par des personnes. On ne peut donc pas les ignorer. Au delà des préférences et des goûts de chacun se pose le problème de la connaissance. On ne devrait pas définir une stratégie de conception et déploiement d'un logiciel sans prendre en compte les personnes qui l'utilisent et l'entretiennent.
- Un utilisateur de logiciel est rarement seul : il communique avec d'autres. La dimension organisationnelle ne peut pas être ignorée. Les logiciels manipulent des informations, l'information est une composante essentielle du fonctionnement des organisations, en particulier des administrations publiques. Modifier les flux d'information a un effet sur la structure de pouvoir et d'autorité, sur les procédures et règlements et sur l'infrastructure (organisation des ressources matérielles) des organisations. Certains logiciels ont un impact limité sur l'information, par exemple un système d'exploitation, d'autres un impact direct comme un logiciel de gestion du temps.
- La durée de vie d'un logiciel, ou du moins d'une version, est en général courte alors que celle de son besoin (les données et documents) est d'autant plus longue et que son contexte d'utilisation est crucial et complexe. Le contexte des NTIC (Nouvelles Technologies de l'Information et de la Communication) est extrêmement évolutif et semble imposer des réactions rapides alors que leur mise en oeuvre et les besoins qu'elle satisfait impliquent souvent une longue période. Il ne faut donc pas limiter l'horizon temporel du raisonnement conduisant à des choix de logiciel à quelques mois, il devrait être de quelques lustres (en fait certainement au moins dix ans dans le contextes des grandes administrations). Ceci pose un problème : on ne doit pas prendre de décision en fonction de l'état du moment mais en fonction de ce que deviendra le contexte de l'utilisation des logiciels, évidemment cela n'est pas facile ; quand on regarde rétrospectivement la qualité des prévisions des gourous, journalistes et conseillers spécialisés, ... on ne peut que frémir et convenir qu'il vaut mieux prendre en compte aussi le contexte que de se limiter à la dimension technologique .

l'installer sur votre disque dur, elle démarre en moins de 90 secondes sur mon ordinateur portable (connexion à Internet comprise).

- L'adoption (incluant l'acquisition) et l'utilisation d'un logiciel entraîne des coûts : la dimension économique ne peut pas être ignorée. Ces coûts concernent non seulement le logiciel, mais aussi le matériel (incluant les réseaux) et, surtout, les personnes nécessaires à leur mise en oeuvre et à leur entretien ainsi bien sûr celles qui utilisent les informations créées, communiquées et conservées par les logiciels. Un logiciel n'agit pas seul, il interagit avec d'autres, ce qui fait que l'évolution de l'un a des conséquences sur les autres, entraînant ainsi une cascade d'effets qui occasionnent des coûts. Ces coûts peuvent être évalués du point de vue de la collectivité des utilisateurs (une organisation), mais aussi à l'échelle individuelle (l'utilisateur) ou globale (la société ou le pays). En effet les conséquences d'une dépense découlent de la dépense. On peut légitimement penser que la dépense de frais de licences demandés par une entreprise étrangère n'a pas le même effet sur une collectivité (un pays) que le paiement de salaires liés à des emplois de qualité à des ressortissants du pays : dans un cas le pays s'appauvrit, dans l'autre il s'enrichit. On peut argumenter que l'usage du logiciel augmente la productivité, mais c'est vrai quelque soit l'origine d'un logiciel, il est question ici de l'effet indirect de la dépense, pas de l'effet direct lié à l'usage du logiciel.

On ne doit pas aborder la problématique des logiciels libres de la même manière que celle des logiciels propriétaires. En particulier le recours aux logiciels libres, que ce soit pour un logiciel d'usage général (par exemple un système d'exploitation ou un logiciel de composition ou de mise en page de texte) ou un logiciel plus spécifique (par exemple un logiciel de gestion d'un dossier médical ou un logiciel de gestion des matériels et de l'activité pédagogique) ne se limite pas à un choix de fournisseur dans le contexte d'un appel d'offres. On peut décider d'étudier l'option logiciel libre comme une alternative, mais cette alternative ne peut pas être considérée simplement comme un des choix de fournisseur possible. On n'est pas client d'un logiciel libre, mais un membre de la communauté des utilisateurs-développeurs du logiciel en question. Bien sûr, il n'est pas question que chaque utilisateur de logiciel libre participe à la création du logiciel ! Si plusieurs dizaines de milliers de personnes du gouvernement utilisaient un logiciel libre de composition de texte, ces personnes ne participeraient pas toutes à la communauté des "producteurs" du logiciel, mais leur employeur le ferait en partie. Tout d'abord il assurerait, à moindre coût, une fonction coûteuse : celle de vendeur. En effet, il n'y a pas de vendeur d'un tel logiciel, l'utilisateur-acquéreur doit se convaincre lui-même de l'avantage d'utiliser ce genre de logiciel, aucun vendeur ne viendra frapper à sa porte. Il doit se charger d'organiser, ou de trouver le service relatif à ce logiciel. Il a intérêt à suivre l'évolution du logiciel et à faire en sorte que ses besoins d'évolution soient pris en compte et éventuellement participer à leur réalisation. Chaque utilisateur n'est pas un membre de la collectivité des développeurs du logiciel mais globalement leur employeur l'est. On ne peut donc pas être un simple client consommateur, on doit être un utilisateur pro-actif, ne serait-ce que dans l'obtention du logiciel : il faut aller le chercher, ne serait-ce qu'en affichant clairement sa volonté de recourir aux logiciels libres, ce qui va provoquer une offre de service incluant le recours à des logiciels libres.

5.5 Éléments pour la construction de scénarios

5.5.1 Les ancrages en ce qui concerne la définition d'une politique face au logiciel libre : les raisons et motivations

Définir une politique en ce qui concerne les logiciels libres est un problème complexe comportant de multiples facettes : technique bien sûr, mais aussi humaine, organisationnelle, économique et politique. Il y a donc plusieurs ancrages⁷⁷ possibles pour élaborer et décider un plan d'action en regard du phénomène des logiciels libres. Un point d'ancrage que l'on rencontre de plus en plus au niveau gouvernemental est : "assurer la création, communication et conservation des informations dans des conditions pérennes, sécuritaires, économiques et dans le respect des droits et devoirs des citoyens et des employés et fournisseurs du gouvernement".

On peut vouloir envisager l'usage des logiciels libres pour plusieurs raisons :

- Le respect des normes et standards publics garantissant l'accès libre et permanent aux informations ;
- La sécurité assurée par le libre accès au code source ;
- La garantie de pérennité assurée par le libre accès au code source ;
- La flexibilité assurée par le libre accès au code source ;
- La qualité (stabilité, sécurité, fonctionnalité, ...);
- Le partage d'expertise ;
- La mise en commun de ressources humaines ;
- Le développement d'expertise locale ;
- La contribution à l'économie locale ;
- Des raisons éthiques ou philosophiques ;
- Des raisons juridiques ;
- Parce que d'autres les utilisent (effet réseau) ;
- Parce que d'autres les utilisent (effet d'entraînement) ;
- Une économie du TCO (coût total de possession) ;
- Une économie sur les licences ;
- Les effets recherchés sur l'économie locale (recours à des services locaux) : un pays s'enrichit en procurant des emplois technologiques à ses citoyens mais s'appauvrit quand il paye des redevances à une firme étrangère.

5.5.2 Par où commencer

La principale difficulté est le coût de transition. Ce coût concerne d'une part les connaissances et l'expertise (ou les habitudes) des utilisateurs des logiciels mais aussi et surtout la conversion des applications et des documents en usage. Or, ce coût semble récurrent ; les grands éditeurs imposent de tels coûts de conversion périodiquement. C'est ainsi que l'on peut légitimement penser que sur une longue période (de l'ordre de dix ans), le passage au logiciel libre permettra de réelles économies de conversion, car le besoin et le rythme de conversion est dicté par les utilisateurs et non les producteurs du logiciel.

⁷⁷Je me réfère ici à la méthode de Peter Checkland : *Systems Thinking, Systems Practice*, 1999, John Wiley & Sons Canada, Ltd., ISBN : 0471986062

Plusieurs stratégies ont été élaborées et pratiquées :

- Décider au sujet des normes et standards d'échange et de conservation. De toute évidence le grand candidat actuel est XML. En attendant l'usage généralisé de XML, la principale option pour les documents est de recourir le plus systématiquement possible au format pdf pour la communication des documents. Ce n'est pas une "bonne" solution, mais elle est souvent citée comme la moins mauvaise. Il est assez dramatique de constater que l'on n'ait pas de solution publique universellement acceptée pour définir la structure des documents échangés par les administrations. Il est toutefois intéressant de remarquer que les logiciels libres sont en avance sur les logiciels propriétaires en ce qui concerne l'usage de XML (même si certaines façons de faire doivent probablement être ajustées, comme par exemple celle actuellement pratiquée par Openoffice).
- Privilégier les serveurs et les logiciels de réseau (pare-feu par exemple). C'est une stratégie dominante. Linux a fait ses preuves dans ce domaine, c'est donc techniquement une stratégie peu risquée. L'impact est limité aux informaticiens, les utilisateurs conservent leurs postes de travail et leurs habitudes.
- Les serveurs d'application, en particulier pour les intranets. Linux, Apache, PHP, OpenLDAP et MySQL⁷⁸ par exemple ont fait leurs preuves. Les environnements de développement sont complets et puissants, et ils sont appuyés par des communautés actives. C'est là aussi une stratégie peu risquée mais qui touche une communauté d'informaticiens plus grande. De plus en plus de solutions sont disponibles pour la réalisation d'intranets robustes et de qualité (Zope par exemple).
- Déployer des applications générales, comme l'accès à Internet (courriel et furetage) ou bureautiques (traitement de textes) : elles peuvent être multi-plate-forme comme OpenOffice ou Mozilla, ce qui facilite leur déploiement. Ces applications n'impliquent pas forcément de recourir à Linux, c'est ainsi que l'on voit certaines administrations (souvent des écoles) utiliser OpenOffice sur Windows. Ceci est spécialement envisageable si l'on décide que les documents échangeables doivent suivre la norme XML et respecter certaines structures tout en gardant une compatibilité acceptable avec la situation précédente.
- Orienter les nouveaux développements d'application maison (cas des refontes).
- Opter pour des postes de travail avec Linux et un environnement de bureau comme KDE dont la convivialité et la stabilité sont de plus en plus de niveau "professionnel", formule techniquement réalisable mais qui implique de nombreux changements de la part des utilisateurs.
- Choisir des applications spécialisées sur des postes de travail dédiés : l'impact sur les utilisateurs est alors limité aux fonctionnalités de l'application (caisse automatique par exemple).

5.5.3 L'action gouvernementale

Plusieurs démarches sont praticables (elles ne sont pas exclusives) :

- Légiférer ou réglementer : de la loi à la directive. Ce devrait être le cas pour les normes. En ce qui concerne les logiciels ceci est discutable. D'un point de vue

⁷⁸LAMP : on a même formé cet acronyme avec Linux, Apache, Mysql et PHP.

juridique, il semble possible pour un gouvernement de légiférer, mais en pratique il est probablement mieux d'exiger de la rigueur dans l'évaluation des projets et de s'assurer que la solution logiciel libre soit une option systématiquement envisagée.

- Encourager : priorités pour l'allocation de ressources humaines locales.
- Promouvoir : informer et diffuser les réussites.
- Agir : projets phares ou pilotes (l'aspect pilote étant plus relatif à la dimension organisationnelle -incluant l'apprentissage- que technique car de nombreux outils libres ont déjà fait leurs preuves ailleurs).
- Gérer : par exemple, exiger la démonstration de l'efficacité des changements et mises à jour.
- Organiser : mettre en place des centres de soutien et organiser la coopération.

5.5.4 L'approche

L'approche dépend en grande partie de la marge de manoeuvre et de la volonté de ses promoteurs :

- planifiée et progressive ;
- libre et progressive ;
- radicale : exemple Linux partout le plus vite possible ;
- mixte : logiciel libre coexistant avec logiciel propriétaire.

5.5.5 Les écueils

On peut relever de nombreux écueils relatifs dans les diverses dimensions de ce phénomène complexe, dont les principaux sont :

- besoin d'un parapluie psychologique ;
- manque de formation des utilisateurs et des informaticiens ;
- résistance au changement ;
- menace des brevets ;
- manque (perception) d'assistance ;
- lobby des propriétaires et la puissance du marketing et des forces de vente ;
- manque de normes et de standards ;
- relation travail / maison-loisir ;
- attirance des gadgets ;
- individualisme ;
- manque de connaissance de l'existence des logiciels libres ;
- attitudes (si c'est pas cher c'est pas bon, si c'est gratuit ça ne vaut rien).

6 En guise de conclusion : quatre questions, quatre réponses

Aux quatre questions principales posées en début de ce document, et qui constituent la raison d'être de cette étude sur le phénomène des logiciels libres, nous répondons ici par des arguments, points clés et options envisageables par un gouvernement.

6.0.6 Le Gouvernement du Québec doit-il envisager de participer au phénomène du logiciel libre : oui !

- Plus de 25 gouvernements, de nombreuses administrations, entreprises et individus sont d'ores et déjà engagés dans le phénomène des logiciels libres.
- Les gains économiques directs et indirects sont démontrés.
- Reprise de la maîtrise de l'évolution des logiciels.
- Les avantages sont concrets (sinon mesurables).
- Les risques techniques sont faibles.
- L'environnement est prêt (du moins de manière latente).
- La communauté des spécialistes est déjà significative et est en forte croissance.

6.0.7 Le Gouvernement du Québec tirera-t-il des avantages à participer au phénomène du logiciel libre : oui !

- Renforcement de l'usage de standards et de normes.
- Respect du citoyen.
- Pérennité et sécurité.
- Enrichissement collectif.
- Diminution de certains coûts.
- Ralentissement de l'obsolescence du matériel.

6.0.8 Le Gouvernement du Québec a-t-il les moyens de participer au phénomène du logiciel libre : oui !

- Les expertises nécessaires ne sont pas toujours différentes de celles qui sont requises par les logiciels propriétaires, elles sont donc disponibles.
- Ce n'est pas une question de budget.
- De toute façon il faudra assumer des coûts de conversion, alors mieux vaut les assumer une bonne fois pour toute pour être ensuite en situation de contrôle (une grande partie de ces coûts sont liés aux normes et standards).

6.0.9 Le Gouvernement du Québec devrait envisager de participer au phénomène du logiciel libre : le plus tôt possible !

- Les logiciels libres sont disponibles et évoluent rapidement, une fonction absente aujourd'hui pourra être disponible demain.
- Les coûts de conversion, il y en a mais ils sont de toute façons récurrents dans le monde du logiciel propriétaire. Plus on attend plus ils seront élevés.

7 Annexes

7.1 La lettre du DR. EDGAR DAVID VILLANUEVA NUÑEZ, Membre du Congrès de la République du Pérou en réponse au Directeur Général de Microsoft Pérou

Parmi les documents explicitant la position d'un gouvernement face aux logiciels libres, cette lettre est exemplaire. Elle a fait le tour du monde (elle est traduite en plusieurs langues par des personnes promouvant les logiciels libres dans leur environnement), elle répond point par point aux principaux arguments des entreprises de logiciels propriétaires dont Microsoft est de toute évidence le principal représentant.

Lima, le 8 avril 2002.⁷⁹
Monsieur JUAN ALBERTO GONZÁLEZ
Directeur Général de Microsoft Pérou
Cher Monsieur.

Avant toute chose, je vous remercie de votre lettre du 25 mars 2002 dans laquelle vous exprimez la position officielle de Microsoft concernant le **Projet de Loi N° 1609, Logiciel Libre dans l'Administration Publique**, qui est inspirée sans aucun doute par le désir d'aider le Pérou à réussir à trouver sa place dans le contexte technologique global. Animé du même esprit et convaincu que nous trouverons les meilleures solutions par l'échange d'idées claires et ouvertes, je me permets de répondre, par la présente, aux commentaires contenus dans votre lettre.

Je reconnais que des opinions comme les vôtres constituent un apport significatif, mais elles m'eussent été plus utiles si, en plus de formuler des objections à caractère général (que nous analyserons en détail plus loin) vous aviez rassemblé des arguments solides sur les avantages que le logiciel propriétaire peut apporter à l'État péruvien et à ses citoyens en général, car cela aurait pu permettre un échange plus clair dans le respect des positions de chacun.

Dans le but de clarifier le débat, nous conviendrons que ce que vous appelez "logiciel à code source ouvert" est ce que le Projet définit comme "logiciel libre", sachant qu'il existe du logiciel dont le code source est distribué avec les programmes, mais qui n'est pas couvert par la définition établie dans le Projet; et que ce que vous appelez "logiciel commercial" est ce que le Projet définit comme "propriétaire" ou "non libre", sachant qu'il existe du logiciel libre commercialisé sur le marché avec un prix comme tout autre bien ou service.

De même il est important de préciser que la proposition contenue dans le Projet auquel nous nous référons n'est pas directement en relation avec l'économie directe qui peut être réalisée par l'emploi de logiciel libre dans les institutions de l'État. Ceci est dans tous les cas, une valeur ajoutée marginale, mais en aucune manière l'objectif

⁷⁹Traduction <http://www.ecole.eu.org/textes/rescon-fr.html>

final du Projet. Les principes élémentaires qui inspirent le Projet sont liés aux garanties fondamentales d'un État démocratique de droit, telles que :

1. Libre accès du citoyen à l'information publique ;
2. Pérennité des données publiques ;
3. Sécurité de l'État et des citoyens.

Pour garantir le libre accès des citoyens à l'information publique, il est indispensable que l'encodage des données ne soit pas lié à un fournisseur unique. L'utilisation de formats standards et ouverts permet de garantir ce libre accès, et d'obtenir, si nécessaire, la création de logiciel libre compatible.

Pour garantir la pérennité des données publiques, il est indispensable que l'utilisation et le maintien du logiciel ne dépendent pas de la bonne volonté des fournisseurs, ni des conditions de monopole imposées par ceux-ci. Pour cela l'État a besoin de systèmes dont l'évolution puisse être garantie par la disponibilité du code source.

Pour garantir la sécurité de l'État ou sécurité nationale, il est indispensable de se baser sur des systèmes dépourvus d'éléments qui en permettent le contrôle à distance ou la transmission non désirée d'information à des tiers. Par conséquent, il faut des systèmes dont le code source est librement accessible au public pour permettre son examen par l'État lui-même, les citoyens, et un grand nombre d'experts indépendants dans le monde. Notre proposition apporte un plus de sécurité, puisque la connaissance du code source élimine le nombre croissant de programmes contenant potentiellement du *code espion*.

De cette façon, notre proposition renforce la sécurité de nos citoyens, à la fois en tant que détenteurs légitimes de l'information gérée par l'État, et en tant que consommateurs. Dans ce dernier cas, c'est en permettant l'apparition d'une offre étendue de logiciel libre dépourvu de potentiel *code espion* susceptible de mettre en péril la vie privée et les libertés individuelles.

En ce sens, le projet de loi se limite à établir les conditions dans lesquelles les organismes de l'État acquerront du logiciel dans le futur, à savoir, de façon compatible avec la garantie de ces principes fondamentaux.

A la lecture du projet il apparaîtra clairement qu'une fois approuvée :

- la loi n'interdit pas la production de logiciel propriétaire ;
- la loi n'interdit pas le commerce de logiciel propriétaire ;
- la loi ne dicte pas quel logiciel utiliser concrètement ;
- la loi ne dicte pas chez quel fournisseur acheter le logiciel ;
- la loi ne limite pas les termes de la licence qui couvre un produit logiciel.

Ce que le projet exprime clairement c'est que, pour être acceptable par l'État, il ne suffit pas que le logiciel soit techniquement suffisant pour mener à bien une tâche, mais il faut en plus que ses conditions contractuelles satisfassent une série de pré-requis en matière de licence, sans lesquelles l'État ne peut pas garantir au citoyen le traitement adéquat de ses données, veiller à leur intégrité, leur confidentialité et leur accessibilité au cours du temps, car ce sont des aspects critiques de son usage normal.

Nous sommes d'accord, Mr. González, sur le fait que la technologie de l'information et des communications a un impact significatif sur la qualité de vie des citoyens (sans que pour eux, l'impact soit toujours positif ou neutre d'effet). De même nous serons certainement d'accord pour dire que les valeurs de base que j'ai signalées plus haut

sont fondamentales dans une nation démocratique comme le Pérou. Depuis longtemps nous cherchons une alternative permettant de garantir ces principes, qui ne consiste pas à recourir à l'emploi de logiciel libre dans les termes définis dans le projet de Loi.

Quant aux observations que vous formulez, nous allons maintenant les examiner dans le détail :

En premier lieu, vous signalez que : *"1. Le projet établit l'obligation pour tout organisme public d'employer exclusivement du logiciel libre, c'est-à-dire à code source ouvert, ce qui transgresse les principes de l'égalité devant la loi, de non-discrimination et les droits à la libre initiative privée, liberté d'entreprendre et de contrat, protégés par la constitution."*

Cette appréciation est une erreur. En aucune façon le projet n'affecte les droits que vous énumérez : il se limite à établir les conditions pour l'emploi de logiciel au sein des institutions de l'État, sans s'immiscer d'aucune manière dans les transactions du secteur privé. C'est un principe bien établi que l'État n'a pas la grande liberté de contrat du secteur privé, précisément parce qu'il est limité dans ses actions par le devoir de transparence des actes publics ; et en ce sens, la préservation de l'intérêt commun doit prévaloir lorsqu'il légifère en la matière.

Le projet protège l'égalité devant la Loi, et aucune personne physique ou morale n'est exclue du droit d'offrir ces biens à l'État dans les conditions fixées dans le projet et sans plus de limitations que celles établies dans la loi des Contrats et Acquisitions de l'État (T.U.O. par Décret Suprême No. 012-2001-PCM).

Le projet n'introduit aucune discrimination, puisqu'il établit uniquement *comment* ces biens doivent être fournis (ce qui est une prérogative d'État) et non *qui* doit les fournir (ce qui serait effectivement discriminatoire si les restrictions étaient fondées sur l'origine nationale, raciale, religieuse, idéologique, la préférence sexuelle, etc.) Au contraire, le projet est résolument anti-discriminatoire. Il en est ainsi parce qu'en déterminant, sans l'ombre d'un doute possible, les conditions de sélection d'un logiciel, il évite aux organismes de l'État d'utiliser des programmes dont la licence inclurait des conditions discriminatoires.

Il résulte de ce qui a été exposé dans les paragraphes précédents, que le projet n'attente pas à la libre initiative privée, puisque celle-ci peut choisir sous quelles conditions elle produit un logiciel ; certaines d'entre elles seront acceptables pour l'État, et d'autres ne le seront pas parce qu'elles contrediraient la garantie des principes fondamentaux énumérés plus haut. Cette libre initiative est compatible avec la liberté d'entreprendre et la liberté de contrat (dans les limites où l'État peut exercer cette dernière). Tout sujet privé peut produire du logiciel selon les conditions requises par l'État, ou peut s'abstenir de le faire. Personne n'est forcé d'adopter un modèle de production, mais si quelqu'un désire fournir du logiciel à l'État, il lui faudra mettre en oeuvre des mécanismes garantissant les principes qui sont décrits dans le projet.

En guise d'exemple : rien dans le texte du projet n'interdit à votre société d'offrir aux organismes de l'État sa "suite" bureautique, dans les conditions définies dans le projet et à un prix que vous jugerez convenable. Si vous ne le faites pas, cela ne sera pas dû à des restrictions imposées par la loi, mais à des décisions de votre société tenant compte du mode de commercialisation de ses produits, décisions auxquelles l'État ne participe pas.

En poursuivant, vous signalez que : *"2. Le projet, en rendant obligatoire l'emploi de logiciel à code source ouvert, établira un traitement discriminatoire et non compétitif pour les contrats et les acquisitions des organismes publics..."*

Cette affirmation est une répétition de la précédente, la réponse se trouve quelques lignes plus haut. Cependant, arrêtons nous un instant sur votre appréciation concernant le "traitement ... non compétitif."

A l'évidence, au moment de définir un quelconque type d'acquisition, l'acheteur se fixe des conditions liées à l'usage prévu pour le bien ou le service. A partir de là, il exclut certains fabricants qui n'auront pas la possibilité de rivaliser, sans pour autant les avoir exclus "a priori", mais sur la base d'une série de principes décidés par la volonté autonome de l'acheteur, si bien que le processus s'avère finalement conforme à la loi. Et dans le projet il est établi que *personne*, n'est exclu de la compétition pour autant que la garantie des principes fondamentaux est satisfaite.

De plus le projet *stimule* la compétition, du moins il pousse à générer une offre de logiciel présentant de meilleures conditions d'utilisation, et à optimiser les travaux déjà accomplis, dans un modèle de progrès continu.

D'un autre côté, l'aspect central de la compétitivité est l'opportunité de proposer de meilleures options au consommateur. Il est impossible d'ignorer que le marketing ne joue pas un rôle neutre au moment de la présentation d'une offre au marché (du moins admettre le contraire reviendrait à dire que les investissements réalisés par les entreprises en matière de marketing sont dépourvus de sens), et par conséquent une dépense significative dans ce domaine peut influencer les décisions de l'acheteur. Cette influence du marketing est dans une large mesure réduite par le projet que nous soutenons, puisque le choix proposé dans le marché se base sur le *mérite technique* du produit et sur l'effort de commercialisation du producteur ; en ce sens, la compétitivité est accentuée, et même le plus petit producteur de logiciel peut rivaliser sur un pied d'égalité avec la plus puissante des entreprises.

Il est nécessaire de souligner qu'il n'y a pas de position plus anti-compétitive que celle des grands producteurs de logiciel propriétaire, qui fréquemment, abusent de leur position dominante, parce que dans d'innombrables cas ils proposent comme unique solution aux problèmes soulevés par les utilisateurs : "mettez à jour vos logiciels vers la nouvelle version" (à la charge de l'utilisateur évidemment) ; de plus, les interruptions arbitraires d'assistance technique sur des produits, jugés "anciens" par le fournisseur, sont communes ; ensuite pour obtenir une quelconque assistance technique, l'utilisateur est contraint de migrer (avec un coût non trivial, en particulier lorsque la migration implique des changements de plate-forme matérielle) vers de nouvelles versions. Et comme toute l'infrastructure est consolidée par des formats de données propriétaires, l'utilisateur reste "captif" de la nécessité de continuer à employer les produits du même fournisseur, à moins de consentir un énorme effort pour passer à un autre environnement (probablement tout aussi propriétaire).

Vous ajoutez : *"3. Ainsi, en obligeant l'État à favoriser un modèle de commerce qui s'appuie exclusivement sur le logiciel à code source ouvert, le projet ne fera que décourager les sociétés de fabrication locales et internationales qui sont celles qui réalisent les véritables investissements, créent un nombre significatif d'emplois directs*

et indirects et contribuent au PIB contrairement à un modèle de logiciel à code source ouvert qui tend à avoir un impact économique toujours moindre du fait qu'il crée principalement des emplois de service."

Je ne suis pas d'accord avec ce que vous affirmez. En partie à cause de ce que vous-même signalez dans le paragraphe 6 de votre lettre, concernant le poids relatif des services dans le contexte de l'utilisation du logiciel. Cette contradiction, par elle-même, invalide votre position. Le modèle des services, adopté par un grand nombre d'entreprises de l'industrie informatique, est bien plus significatif, en termes économiques, et de façon croissante, que le commerce de licences sur les programmes.

D'un autre côté, le secteur privé dispose de la plus grande liberté pour choisir le modèle économique qui convient le mieux à ses intérêts, même si cette liberté de choix est souvent obscurcie de manière subliminale par les investissements disproportionnés dans le marketing des producteurs de logiciel propriétaire.

De plus, à la lecture de votre opinion il ressort que le marché de l'État est crucial et indispensable pour l'industrie du logiciel propriétaire, à tel point que si l'État adopte ce projet, il éliminerait complètement ces sociétés du marché. En supposant, ce qui n'est pas le cas, que ce soit vrai, nous en déduisons que l'État subventionne l'industrie du logiciel propriétaire. Dans cette hypothèse peu probable, l'État aurait alors le droit d'attribuer ses subventions au domaine qu'il considère comme ayant la plus grande valeur sociale il en résulterait que si l'État décide de subventionner le logiciel il devra le faire en préférant le libre par rapport au propriétaire, compte tenu de son effet social et de son utilisation rationnelle de l'argent des contribuables.

Concernant les emplois générés par le logiciel propriétaire dans des pays comme le nôtre, ceux-ci concernent majoritairement des tâches techniques de faible valeur ajoutée ; au niveau local, les techniciens qui offrent du support au logiciel propriétaire produit par des entreprises transnationales ne sont pas en mesure de corriger un bug, pas nécessairement faute de capacité technique ou de talent, mais parce qu'ils ne disposent pas du code source. Le logiciel libre crée des emplois techniquement plus qualifiés et on génère un cadre pour la libre concurrence où le succès n'est limité que par la capacité d'offrir du bon support technique et de la qualité de service, on stimule le marché et on enrichit le patrimoine commun de la connaissance, en ouvrant des alternatives pour générer des services de grande valeur ajoutée et de meilleur profil de qualité profitant à tous les acteurs : producteurs, prestataires de services et consommateurs.

C'est un phénomène courant dans les pays en voie de développement que les industries locales de logiciel tirent la majeure partie de leurs revenus des services ou de la fabrication de logiciel "ad hoc". Par conséquent, l'éventuel impact négatif que l'application du projet pourrait avoir dans ce secteur sera compensé par la croissance de la demande de services (à condition que ceux-ci soient conformes aux exigences de qualité). Évidemment, il est probable que les entreprises transnationales de logiciel décidant de ne pas concourir conformément à ces règles du jeu, souffrent d'une perte de revenus en termes de facturation de licences ; néanmoins, considérant que ces entreprises soutiennent que beaucoup de logiciels utilisés par l'État ont été copiés illégalement, on peut penser que l'impact ne sera pas très sérieux. Certainement, en tout cas, leur succès sera déterminé par les lois du marché dont les changements ne peuvent être évités ; de nombreuses entreprises traditionnellement associées au logiciel propriétaire ont déjà franchi le pas (au prix d'investissements importants) pour offrir des services

associés au logiciel libre, ce qui démontre que les modèles ne sont pas mutuellement exclusifs.

Avec ce projet l'État décide de préserver certaines valeurs fondamentales. Et il le décide sur la base de ses pouvoirs souverains, sans affecter par là aucune des garanties constitutionnelles. Si ces valeurs peuvent être garanties sans avoir à choisir un modèle économique donné, les effets de la loi seront plus bénéfiques encore. En tout cas, il doit rester clair que l'État n'opte pas pour un modèle économique ; s'il s'avérait qu'il n'existe qu'un seul modèle économique capable de fournir du logiciel qui satisfasse la garantie de base de ces principes, cela relèverait de circonstances historiques et non d'une décision arbitraire en faveur d'un modèle donné.

Poursuivant votre lettre : *"4. Le projet de loi impose l'utilisation de logiciel à code source ouvert sans considérer les dangers que ceci peut entraîner d'un point de vue de la sécurité, de la garantie et des possibles violations des droits de propriété intellectuelle de tiers."*

Faire allusion de façon abstraite aux "dangers que ceci peut entraîner", sans spécifier un seul exemple de ces supposés dangers, dénote une méconnaissance du sujet. Aussi, permettez-moi d'illustrer quelques-uns de ces points.

Concernant la sécurité :

La sécurité nationale a déjà été évoquée dans les principes fondamentaux du projet. En termes plus précis concernant la sécurité du logiciel lui-même, il est bien connu que le logiciel (propriétaire ou libre) contient des erreurs de programmation ou "bugs" (en jargon informatique) dans ses lignes de code. De même, il est de notoriété publique que les bugs dans le logiciel libre sont moins nombreux, et qu'ils sont réparés bien plus rapidement, que dans le logiciel propriétaire. Ce n'est pas en vain que de nombreux organismes publics responsables de la sécurité informatique des systèmes d'institutions de l'État dans les pays développés recommandent l'utilisation de logiciel libre dans des conditions égales de sécurité et d'efficacité.

Il est impossible de prouver que le logiciel propriétaire est plus sûr que le libre, sauf par un examen détaillé, public et ouvert, par la communauté scientifique et les utilisateurs en général. Or, cette démonstration est impossible parce que le modèle même du logiciel propriétaire interdit cette analyse, si bien que la garantie de sécurité se base sur la parole ambiguë (mais vraisemblablement partielle) du producteur du logiciel ou de ses contractants.

Il faut se souvenir que, dans de nombreux cas, les conditions de la licence incluent des clauses de confidentialité [NdT : « NDA » ou non disclosure agreement] qui interdisent aux utilisateurs de révéler ouvertement les failles de sécurité découvertes dans le produit propriétaire sous licence.

Respect de la garantie :

Comme vous le savez parfaitement, ou pourrez le découvrir en lisant le "Contrat de Licence pour l'Utilisateur Final" [NdT : EULA] des produits dont vous commercialisez la licence, dans la très large majorité des cas, les garanties sont limitées au remplacement du support de distribution s'il est défectueux, mais en aucun cas elles ne prévoient de compensations pour les dommages directs ou indirects, manque à gagner, etc. si suite à un bug de sécurité dans un quelconque de vos produits, non réparé par vous, un attaquant parvenait à compromettre des systèmes cruciaux pour les services de

l'État : quelle garantie, quelles réparations ou quelles compensations donneraient votre société en accord avec les conditions de votre licence ? Les garanties du logiciel propriétaire, comme les programmes sont livrés "AS IS" [NdT : tel quel], ce qui veut dire dans l'état dans lequel ils se trouvent, sans aucune responsabilité additionnelle pour le fournisseur concernant sa fonctionnalité, ne diffèrent aucunement de celles habituelles dans le logiciel libre.

Sur la propriété intellectuelle :

Les questions de propriété intellectuelle dépassent le cadre de ce projet, et elles sont couvertes par d'autres lois spécifiques. Le modèle du logiciel libre n'implique en aucune façon l'ignorance de ces lois et en fait, la grande majorité du logiciel libre est couverte par le copyright. En réalité, la seule présence de cette question dans vos observations démontre votre confusion quant au cadre légal où vit le logiciel libre. L'incorporation de la propriété intellectuelle d'autrui dans des travaux que l'on s'attribue par la suite n'est pas une pratique courante de la communauté du logiciel libre ; en revanche, c'est malheureusement le cas sur le terrain du logiciel propriétaire. Prenez comme exemple la condamnation par le Tribunal de Commerce de Nanterre, France, le 27 septembre 2001, de Microsoft Corp., à 3 millions de francs en dommages et intérêts, pour violation de la propriété intellectuelle (piratage, pour utiliser le terme malheureux que votre société utilise couramment dans ses publicités).

Vous poursuivez en disant que : "5. Le projet utilise de manière erronée les concepts du logiciel à code source ouvert, qui n'est pas nécessairement du logiciel libre ou de coût nul, aboutissant à des conclusions équivoques sur les économies pour l'État, sans une analyse des coûts et bénéfiques pour étayer votre position."

Cette remarque est fautive, en principe la gratuité et la liberté sont des concepts orthogonaux : il y a du logiciel propriétaire et onéreux (par exemple, MS Office), du logiciel propriétaire et gratuit (MS Internet Explorer), du logiciel libre et onéreux (distributions RedHat, SuSE, etc. du système GNU/Linux), du logiciel libre et gratuit (Apache, OpenOffice, Mozilla), et du logiciel sous différentes modalités de licence (MySQL).

Il est certain que le logiciel libre n'est pas nécessairement gratuit. Et le texte du projet ne dit pas qu'il doit l'être comme vous l'aurez bien noté après l'avoir lu. Les définitions incluses dans le projet déterminent clairement *quoi* considérer comme logiciel libre, sans jamais faire référence à la gratuité. Bien qu'il soit fait mention des économies réalisées en terme de non-paiement des licences de logiciel propriétaire, les fondements du projet mentionnent clairement les garanties fondamentales qui doivent être préservées et la stimulation du développement technologique local. Sachant qu'un État démocratique doit respecter ces principes, il ne lui reste aucune autre solution que d'employer du logiciel dont le code source est publiquement disponible et d'échanger de l'information uniquement dans des formats standards.

Si l'État n'employait pas de logiciel présentant ces caractéristiques, il violerait les principes républicains fondamentaux. Par chance, le logiciel libre implique en plus un coût global moindre ; néanmoins, dans l'hypothèse (aisément réfutée) où il coûterait plus cher que le logiciel propriétaire, la seule existence d'un outil logiciel libre efficace pour une fonction informatique déterminée obligerait l'État à l'utiliser ; non par force de ce projet de Loi, mais pour les principes élémentaires que nous avons énumérés au début et qui émanent de l'essence même de l'État de droit démocratique.

Vous poursuivez : *"6. Il est faux de penser que le logiciel à code source ouvert est gratuit. Des études du Gartner Group (organisme étudiant le marché technologique reconnu au niveau mondial) ont révélé que le coût d'acquisition du logiciel (système d'exploitation et applications) ne représente que 8% du coût total que les entreprises et les institutions doivent assumer pour une utilisation rationnelle et réellement bénéfique de la technologie. Les autres 92% sont constitués des coûts d'installation, de déploiement, de support, de maintenance, d'administration et d'indisponibilité."*

Cet argument répète celui déjà donné au paragraphe 5 et contredit en partie le paragraphe 3. Aussi nous nous en remettons aux précédents commentaires à des fins de brièveté. Nonobstant, permettez-moi de signaler que votre conclusion est fautive d'un point de vue logique : que le coût du logiciel selon le Gartner Group ne soit que de 8% du coût total d'utilisation, n'invalide d'aucune manière l'existence de logiciel gratuit, c'est-à-dire, dont le coût de la licence est zéro.

De plus dans ce paragraphe vous indiquez fort justement que les composants de service et les pertes pour indisponibilité forment une partie substantielle du coût total d'utilisation du logiciel ; ce qui, vous le noterez, entre en contradiction avec votre affirmation de la valeur mineure des services suggérée dans le paragraphe 3. En réalité, l'utilisation de logiciel libre contribue significativement à la diminution des coûts restants du cycle de vie du logiciel. Cette réduction de l'impact économique de l'installation, du support, etc. se note dans de nombreux domaines ; d'un côté, le modèle compétitif de services autour du logiciel libre, dont il est possible d'acheter le support et la maintenance auprès d'une offre variée qui rivalise sur le rapport qualité/prix. Ceci est valable pour l'installation, le déploiement, et le support, et en grande partie pour la maintenance. En second lieu, la caractéristique de reproductibilité du modèle fait que la maintenance effectuée pour une application est facilement réutilisable, sans impliquer des coûts importants (c'est-à-dire, sans payer plus d'une fois pour la même chose) car les modifications, si on le souhaite, peuvent être incorporées au patrimoine commun de la connaissance. Troisièmement, l'énorme coût d'indisponibilité ("écrans bleus de la mort", code mal-intentionné tel que les virus, les vers et les chevaux de Troie, exceptions, fautes générales de protection et nombre d'autres maux connus) est considérablement réduit par l'emploi de logiciel plus stable ; et il est bien connu qu'une des vertus les plus remarquables du logiciel libre est sa stabilité.

Vous affirmez plus loin que : *"7. L'un des arguments derrière le projet de loi est la prétendue gratuité du logiciel à code source ouvert, comparée au coût du logiciel commercial, sans tenir compte qu'il existe des modalités de licence en volume qui peuvent être très avantageuses pour l'État, comme cela se fait dans d'autres pays."*

J'ai déjà indiqué que ce qui est en question n'est pas le coût du logiciel, mais les principes de liberté d'information, d'accessibilité et de sécurité. Ces arguments ont été largement traités dans les paragraphes précédents, auxquels je vous prie de vous référer.

D'autre part, il existe certainement des modalités de licence en volume (malheureusement, le logiciel propriétaire ne satisfait pas les principes de base). Mais, comme vous l'avez noté dans le paragraphe immédiatement antérieur de votre lettre, cela ne permet que de réduire l'impact d'un composant qui ne pèse pas plus de 8% du coût total.

Vous poursuivez : *"8. De plus, l'alternative adoptée pour le projet (i) est clairement plus coûteuse du fait des coûts élevés de la migration logicielle, et (ii) met en péril la compatibilité et la possibilité d'interopérabilité des plates-formes informatiques au sein de l'État, et entre l'État et le secteur privé, compte tenu des centaines de versions de logiciel à code source ouvert sur le marché."*

Analysons votre affirmation en deux parties. Le premier argument, celui de la migration qui implique des coûts élevés, est en fait un argument en faveur du projet. En effet, plus le temps passe et plus la migration vers une autre technologie sera onéreuse ; et dans le même temps, les risques de sécurité associés au logiciel propriétaire augmenteront aussi. De cette manière, l'utilisation de systèmes et de formats propriétaires rendra l'État encore plus dépendant des fournisseurs. Au contraire une fois implantée la politique d'utilisation du logiciel libre (implantation qui, certes, a un coût), la migration d'un système vers un autre se fait facilement, puisque toutes les données sont stockées dans des formats ouverts. D'autre part, la migration vers un environnement de logiciel ouvert n'implique pas plus de coûts que celle entre deux environnements distincts de logiciel propriétaire, ce qui invalide complètement votre argument.

Le second argument se réfère à *"l'interopérabilité des plates-formes informatiques au sein de l'État, et entre l'État et le secteur privé"*. Cette affirmation démontre une ignorance des mécanismes de fabrication du logiciel libre, qui ne maximise pas la dépendance de l'utilisateur par rapport à une plate-forme donnée, comme c'est habituellement le cas dans le domaine du logiciel propriétaire. Même lorsqu'il existe plusieurs distributions d'un logiciel libre et plusieurs programmes susceptibles d'être employés pour une même fonction, l'interopérabilité reste garantie autant par l'emploi de formats standards, exigé dans le projet, que par la possibilité de créer un logiciel interopérable à partir du code source disponible.

Vous dites plus loin que : *"9. La majeure partie du logiciel à code source ouvert n'offre pas de niveaux de service adéquats, pas plus que de garantie de fabricants reconnus pour favoriser une grande productivité de la part des utilisateurs, ce qui a conduit différentes organisations publiques à revenir sur leur décision d'utiliser du logiciel à code source ouvert et à utiliser du logiciel commercial en lieu et place."*

Cette observation n'est pas fondée. Compte tenu de la garantie, votre argument est réfuté par la réponse au paragraphe 4. Concernant les services de support, il est possible d'utiliser du logiciel libre sans eux (de la même manière qu'on le fait avec du logiciel propriétaire), mais quiconque le souhaite peut obtenir du support séparément, soit de la part d'une entreprise locale, soit de sociétés internationales, de la même manière que pour le logiciel propriétaire.

D'autre part, vous contribueriez beaucoup à notre analyse si vous pouviez nous donner des informations concernant les projets de logiciel libre *implantés* dans des entités publiques et qui ont été abandonnés en faveur de logiciel propriétaire. Nous connaissons un bon nombre de cas où l'inverse s'est produit, mais n'avons pas d'information au sujet des cas auxquels vous faites référence.

Vous continuez en observant que : *"10. Le projet décourage la créativité de l'industrie péruvienne du logiciel, qui a un chiffre d'affaires de 40 millions de dollars US"*

par an, exporte pour 4 millions de dollars US (10e au rang des produits d'exportation non traditionnels, plus que l'artisanat) et est une source d'emplois hautement qualifiés. Avec une loi qui incite à l'utilisation du logiciel à code source ouvert, les programmeurs de logiciel perdent leurs droits de propriété intellectuelle et leur principale source de revenus."

Il est assez clair que personne n'est obligé de commercialiser son code sous forme de logiciel libre. La seule chose à prendre en compte est que, si cela n'est pas fait, on ne pourra pas le vendre au secteur public. Ce n'est en aucun cas le principal marché pour l'industrie nationale du logiciel. Plus haut nous avons abordé quelques-unes des questions relatives à l'influence du projet sur la génération d'emplois techniques hautement qualifiés et dans de meilleures conditions de compétitivité, il n'est donc pas nécessaire d'insister sur ce point.

Ce qui suit dans votre affirmation est erroné. D'un côté, aucun auteur de logiciel libre ne perd ses droits de propriété intellectuelle, à moins qu'il n'ait exprimé sa volonté de placer son oeuvre dans le domaine public. Le mouvement du logiciel libre a toujours été extrêmement respectueux de la propriété intellectuelle, et a donné une reconnaissance publique très large à ses auteurs. Des noms tels que ceux de Richard Stallman, Linus Torvalds, Guido van Rossum, Larry Wall, Miguel de Icaza, Andrew Tridgell, Theo de Raadt, Andrea Arcangeli, Bruce Perens, Darren Reed, Alan Cox, Eric Raymond, et bien d'autres, sont mondialement reconnus pour leurs contributions au développement de logiciel aujourd'hui utilisé par des millions de personnes partout dans le monde, alors que les noms des auteurs d'excellents composants logiciels propriétaires, demeurent dans l'anonymat. D'un autre côté, affirmer que les revenus de droits d'auteur constituent la source principale de revenus des programmeurs péruviens est pour le moins risqué, en particulier quand on n'a apporté aucune preuve à cet effet, ni aucune démonstration de comment l'emploi de logiciel libre par l'État influencerait ces revenus.

Vous poursuivez en disant que : "11. Le logiciel à code source ouvert, puisqu'il peut être distribué gratuitement, ne permet pas de générer des revenus pour ses développeurs par le biais de l'exportation. De cette manière, on affaiblit la synergie de la vente de logiciel à d'autres pays et par conséquent la croissance de cette industrie, alors qu'au contraire les normes d'un gouvernement doivent stimuler l'industrie locale."

Cette affirmation démontre une fois de plus une méconnaissance totale des mécanismes et du marché du logiciel libre. Elle tente d'affirmer que le marché de cession des droits non exclusifs d'utilisation à titre onéreux (vente de licence) est le seul possible pour l'industrie informatique alors que, comme vous l'avez signalé quelques paragraphes plus haut, il n'est en aucun cas le plus important. Les incitations, émanant de ce projet, à une meilleure offre de personnels qualifiés et à une expérience du logiciel libre à grande échelle permettront aux techniciens nationaux de se placer à un niveau très compétitif sur le marché du travail international.

Vous signalez plus loin que : "12. Dans le Forum on a discuté de l'importance de l'emploi de logiciel à code source ouvert dans l'éducation, sans commentaire sur le retentissant fracas de cette initiative dans un pays comme le Mexique, où précisément les

fonctionnaires de l'État qui fondèrent le projet, déclarent aujourd'hui que le logiciel à code source ouvert ne permet pas d'offrir une expérience d'apprentissage aux écoliers, qu'il n'a pas eu la capacité au niveau national de fournir du support pour cette plate-forme, et qu'il n'a pas pris en compte l'intégration de la plate-forme existante dans les écoles."

Effectivement, le Mexique a fait marche arrière avec le projet Red Escolar. Cela est dû, précisément au fait que les initiateurs du projet mexicain utilisèrent le coût des licences comme principal argument, au lieu des autres raisons stipulées dans notre projet et qui sont plus fondamentales. Compte tenu de cette erreur conceptuelle, aggravée par l'absence d'appui effectif de la part du SEP (Secrétariat à l'Education Publique), ils décidèrent que l'implantation de logiciel libre dans les écoles consistait à suspendre le budget logiciel et en échange à leur envoyer un CD ROM contenant GNU/Linux. Bien sûr, ceci échoua et il ne pouvait en être autrement, de même qu'échouent les laboratoires scolaires qui utilisent des logiciels propriétaires sans disposer d'un budget pour l'installation et la maintenance. C'est précisément pour cela que notre projet de loi ne se limite pas à recommander l'emploi de logiciel libre, mais reconnaît la nécessité et ordonne la création d'un plan de migration viable, dans lequel l'État encadre précisément la transition technique pour bénéficier des avantages du logiciel libre.

Vous terminez par une question rhétorique : "13. Si le logiciel à code source ouvert satisfait tous les pré-requis des entités de l'État pourquoi une loi pour l'adopter ? Ne devrait-ce pas être le marché qui décide librement quels sont les produits qui donnent le plus de bénéfices ou de valeur ?".

Nous sommes d'accord sur le fait que pour le secteur privé, c'est le marché qui doit décider quel produit utiliser et il ne serait pas admissible que l'État interfère. Mais dans le secteur public, le raisonnement n'est pas le même : comme nous l'avons déjà dit, l'État collecte, manipule et transforme de l'information qui ne lui appartient pas, mais qui lui a été confiée par les citoyens qui, par force de loi, n'ont pas d'autre choix que de le faire. En contrepartie de cette obligation légale, l'État doit mettre en oeuvre des mesures extrêmes pour sauvegarder l'intégrité, la confidentialité et l'accessibilité de ces informations. L'emploi de logiciel propriétaire soulève de sérieux doutes quant à l'accomplissement de ces missions, faute d'évidence concluante à ce propos, et par conséquent il n'est pas apte à être utilisé dans le secteur public.

La nécessité d'une loi se fonde d'un côté sur la matérialisation des principes fondamentaux énoncés plus haut dans le domaine spécifique du logiciel ; d'un autre côté, il est un fait que l'État n'est pas une entité idéale homogène mais qu'il est composé de multiples organismes avec différents degrés d'autonomie de décision. Étant donné que l'emploi de logiciel propriétaire est inapproprié, le fait d'établir ces règles dans la loi évitera que la décision discrétionnaire d'un quelconque fonctionnaire mette en péril l'information qui appartient aux citoyens. Et, par-dessus tout, elle constitue une réaffirmation actualisée par rapport aux moyens de traitement et de communication de l'information employés aujourd'hui, du principe républicain du service public.

Conformément à ce principe universellement accepté, le citoyen a le droit de connaître toute l'information en possession de l'État qui ne soit pas couverte par une déclaration de secret conforme à la loi. Le logiciel traite de l'information et il est lui-même de l'information. Information dans un format spécial, susceptible d'être interprété par une

machine pour exécuter des actions, mais sans l'ombre d'un doute information cruciale parce que le citoyen dispose d'un droit légitime de savoir, par exemple, comment se comptabilise son vote ou se calculent ses impôts. Et pour cela, il faut pouvoir accéder librement au code source et éprouver les programmes utilisés pour le comptage électoral ou le calcul des impôts.

Je vous salue avec l'expression de ma considération la meilleure, soyez assuré que mon bureau sera toujours ouvert à l'exposé de vos points de vue, à quelque niveau de détail que vous jugeriez convenable.

Veuillez agréer mes salutations distinguées,

DR. EDGAR DAVID VILLANUEVA NUÑEZ

Membre du Congrès de la République du Pérou.

Traduction et adaptation : Guy Brand <guybrand @ chimie.u-strasbg.fr>

Relectures et corrections : Cyril Chaboisseau, Georges Khaznadar,

Yves Ouvrard, Alain Riffart, Stéphane Casset.

Version originale disponible sur le site de Peruvian Activism.

7.2 Témoignage

La lecture de ce texte, rédigé dans le contexte de cette étude, a déclenché chez plusieurs personnes leur volonté d'explorer à leur tour le monde des logiciels libres. Leur réaction m'a permis de croire qu'il pourrait compléter ce document par une touche plus personnelle et concrète.

J'ai commencé à m'intéresser aux logiciels libres lorsqu'un programmeur de l'équipe de Silverrun m'a parlé de Linux, en 1995. À ce moment là Linux n'était pas encore pour moi, trop difficile à installer, pour le "non-programmeur" que je suis, et pas assez d'applications disponibles. J'étais alors un utilisateur de Mac frustré. En effet, je devais à ce moment là commencer à abandonner mon Mac sous la pression de mon environnement de plus en plus limité à Windows et MS-Office. Je fus l'heureux propriétaire du deuxième Mac livré à Québec (parmi les 50 000 premiers Macs livrés au monde). Auparavant, je possédais alors un Apple II avec lequel j'avais commencé à travailler plutôt que de simplement m'en servir comme un champ d'exploration et de loisir : j'utilisais Gutemberg pour le traitement de texte, Visicalc et Visischedule ainsi que quelques bricolages en Basic et Pascal UCSD ; ces logiciels m'avaient déjà coûté plus de la moitié du prix de ma machine (soit en tout une facture de plus de 4 000 \$ en 1981). J'ai immédiatement apprécié la convivialité du Mac en écrivant mes documents avec Mac Write. J'ai alors assisté de près au débat entre les tenants du Mac et les fanatiques du PC et j'ai pu observer que lors des réunions d'utilisateurs de Mac on discutait de l'usage de notre machine alors que dans les réunions d'utilisateurs de PC on discutait de l'installation des cartes et des logiciels (et accessoirement on copiait des logiciels). J'ai ainsi pu apprécier la logique de ceux (il y en a plus d'un) qui aujourd'hui ne jurent que par Windows après avoir dénigré le système d'exploitation du Mac et en vantant la supériorité de MS-Dos ! En fait, j'ai compris que leur logique n'est pas technologique quoiqu'ils en disent ; ils étaient alors sérieux eux, et ne jouaient pas avec des petites images baptisées icônes et ce truc bizarre qu'est une souris.

Depuis mon premier Mac, j'ai changé plusieurs fois de machine (aux quatre ans en moyenne pour les macs et aux deux ans pour les PC) pour finir la lignée des Macs avec un Powerbook, que j'aimais beaucoup (à l'aise avec le disque de 250 megs du Mac, j'ai découvert que j'étais à l'étroit avec les 750 megs de ma nouvelle machine sous W95). Il est intéressant de noter que si l'on excepte l'usage d'Internet (dont je fais un usage assez intensif maintenant : courriel et information ou documentation sur le Web) et celui d'un outil de modélisation comme Silverrun, les usages que je fais de mon ordinateur personnel ont peu évolué dans leur nature depuis mon premier Apple II (de 64 K de mémoire centrale ! ma machine a aujourd'hui 512 megs) : rédiger des textes (dans lesquels il y a souvent des figures, ce qui était devenu un plaisir avec mes Mac), de plus en plus composer des présentations, et gérer quelques tableaux de calcul ... mais je suis en train de composer ce texte pendant que le CD dans le lecteur me distille du Mozart (eh oui avec uniquement des logiciels libres). Il y a donc quelques agréments au progrès !

À la suite de cette première prise de connaissance de Linux et des logiciels libres, je tombais de temps en temps au hasard de mes lectures sur des nouvelles qui m'intriguaient de plus en plus jusqu'à ce que le PC sous Windows 95 que je gardais à la maison comme ordinateurs de secours en remplacement de mon portable utilisé quo-

tidiennement refuse de démarrer. J'ai réinstallé W95 avec les logiciels livrés (système d'exploitation et CD ou disquette des "drivers") avec la machine et tout marchait mal : pas facile ! Je décidais alors qu'il était temps de jeter un coup d'oeil aux logiciels libres en commençant par Linux. J'ai acheté un livre et commandé la revue LMB no 74 (mai/juin 1998) éditée par le CNRS-ENSTA qui était livrée avec un CD d'installation de Linux (basé sur une distribution Red Hat 5.0) avec des logiciels qui me semblaient être ceux dont j'avais besoin. Tout seul dans mon coin je cherchais une compilation (je ne connaissais pas encore le concept de distribution) qui me semblait être assemblée pour un béotien comme moi. Eh bien ! il a été plus facile de charger Linux que W95 ! Bien sûr j'ai buté sur le modem, j'avais un WinModem que je n'ai pu pas faire fonctionner, je ne comprenais pas pourquoi, il a fallu qu'on m'explique. Cependant, après un ou deux essais et quelques tâtonnements Linux fonctionnait, je pouvais utiliser plusieurs bureaux et j'avais accès à de nombreux et intéressants logiciels. Ce beau dimanche pluvieux là j'ai appris qu'il ne faut pas croire les discours de certains soi-disant spécialistes : il est plus facile d'installer Linux que Windows, à moins bien sûr de recharger une copie de sécurité déjà configurée (j'ai pu vérifier cela plusieurs fois depuis). Par contre, le manque d'accès à Internet (il m'aurait fallu acheter un vrai modem) et surtout le manque de compatibilité avec les documents que je devais réaliser avec MS-Office sur mon portable « obligatoirement » équipé de Windows et MS-office m'ont limité. J'ai donc amené ma machine au service de support technique de ma faculté pour faire ré-installer Windows. Ce jour-là j'ai appris que quand on file droit on peut se faire aider par son support technique, mais que les déviants, eh bien ils ont ce qu'ils méritent ! Qu'ils se débrouillent (je comprends qu'il y a des raisons à ça : en particulier la connaissance et l'inter-opérabilité). Mais malgré cela, j'ai un peu la tête dure, fort de ma position universitaire, je cultive ma déviance tout en espérant qu'un jour elle ne sera plus déviance mais raison largement partagée.

Quelques temps plus tard j'ai appris qu'une des personnes au support technique chez Silverrun, où je travaillais de temps en temps, utilisait régulièrement Linux chez elle. Cela a été pour moi l'occasion de tenter de configurer mon portable en "dual boot" avec Linux. Cae me semblait hors de ma portée et trop risqué : je ne pouvais pas facilement me passer de mon outil de travail. Je fis une sauvegarde du contenu essentiel de ma machine, optais pour "une Mandrake 6" que j'ai télé-chargée avec son aide. Mais cette fois, heureusement que j'ai eu de l'aide pour installer Linux sur mon IBM Thinkpad 390. Ce ne fut pas facile : difficulté à configurer la carte réseau et le moniteur, on n'a pas réussi à configurer le son, mais je n'ai pas trop insisté. Par contre j'ai facilement réussi tout seul l'installation en "dual boot" sur le PC de la maison (j'ai alors découvert que Linux voyait les 13 gigs du deuxième disque dur que je venais d'installer, mais pas Windows ; je ne vous assommerai pas avec les mystères du BIOS et autres réjouissances telles que le partitionnement des disques durs, que j'ai alors du tenter de comprendre partiellement, ainsi que des subtilités du démarrage d'une machine avec Windows ou Linux).

J'avais non pas peur, mais une certaine réticence pour la ligne de commande. Mais le problème n'était pas là : j'avais commencé à comprendre que si pour un utilisateur le bureau graphique est plus plaisant que les commandes, par contre une configuration à base de fichiers texte présentait un grand avantage par rapport à la base de registre de Windows (probablement responsable d'une bonne partie de mes déboires avec W95 ou

W98 affligés d'une instabilité progressive suivie de la mort du système ; un technicien de ma faculté m'a alors expliqué que mon système s'évanouissait parce que je m'en servais trop, il n'est pas bon de charger et décharger de nombreux programmes !). J'étais de plus en plus intéressé par l'environnement Linux et ses bureaux en pleine évolution, mais je butais sur la compatibilité des documents avec MS-Office. Ainsi je commençais à explorer Linux et découvrir le monde des logiciels libres le soir et je travaillais la plupart du temps avec Windows durant le jour.

De version en version, les logiciels de la distribution Mandrake (auquel je suis resté fidèle) me donnaient de plus en plus de satisfaction et les versions devenaient à ma portée : je pouvais les installer tout seul sur mon portable, quitte à demander un peu d'aide pour le figlage. Ce fut assez facile avec le thinkpad 600 E (qui a vu aussi passer W98, NT puis de nouveau W98 en "dual boot") puis le thinkpad A22m (livré initialement avec W2000) avec les versions 7.0, 7.1, 8.0, 8.1, 8.2 et 9.0 de Mandrake (cette dernière est celle que j'utilise actuellement avec le thinkpad A22). Ce faisant je ne respectait pas une des règles de base que j'ai découvert progressivement : ne pas changer quelque chose qui donne satisfaction. Je devenais de plus en plus autonome et fonctionnel sous Linux, surtout depuis la version 6 de StarOffice ou de OpenOffice 1 (malgré mes efforts et mes a-priori positifs, je n'ai pas pu m'adapter à la version 5.2 de StarOffice).

Ainsi, au cours de l'année 2000, Windows a disparu de ma vieille machine à la maison. Elle a été rechargée depuis avec une distribution Debian par des assistants et sert comme serveur pour le projet d'étude des logiciels libres réalisé à l'été 2002 ; je suis en train d'y installer mon site « datarun ». Comme quoi les machines se périment moins vite avec Linux qu'avec Windows, un technicien de ma faculté m'a bien fait comprendre qu'il n'est pas sérieux de monter un serveur NT ou W2000 sur ce vieux pentium MMX 200 (qui m'avait tout de même coûté 4 000 \$ 5 ans plus tôt), même si j'avais augmenté sa mémoire à 96 megs.

Mon portable de travail n'a plus démarré avec W2000 depuis que j'ai pu me faire aider pour résoudre les quelques problèmes que voici (depuis un an il ne démarrait avec W2000 que de temps en temps, quelques fois par mois seulement) :

- Accès aux imprimantes du réseau Novell de ma faculté : le plus difficile, je n'y suis pas arrivé tout seul malgré l'aide de l'assistance de Mandrake. Merci à Nicolas Sylvain qui a fini par configurer ma machine (et documenté cette installation). La difficulté vient de la politique de fermeture de Novell.
- Utilisation de la sortie vidéo pour la projection en cours. Après bien des tâtonnements et des recherches ainsi quelques échanges "sur la toile", j'ai fini par comprendre qu'il fallait que je re-configurais mon moniteur pour avoir une compatibilité de définition. Je pouvais avoir alors l'usage de clés de basculement de l'ordinateur. Cela a été rendu difficile à cause d'un mauvais fonctionnement des outils de configuration vidéo de Mandrake (le point noir pour moi, ma solution est fonctionnelle mais un peu laborieuse) et le fait que j'utilise une haute définition (1400 x 1050. pixels).
- Installation du scanner (Snapscan e52), là encore merci à Nicolas Sylvain, et à la communauté Linux. Il fonctionne maintenant mieux qu'avec le logiciel livré pour Windows (j'utilise Xsane). Depuis, la dernière version de Xsane a résolu ce problème d'installation.

- La liaison avec mon Zaurus (pourtant sous Linux, mais seule la liaison avec Windows était livrée avec cet ordinateur de poche !).
- J’ai aussi parfois eu besoin d’aide pour résoudre certains conflits ou manque de bibliothèques lors de l’installation de certains logiciels (comme la version 1.2 de Lyx avec la distribution Mandrake 8.2, par contre elle s’est installée sans problème avec Mandrake 9.0).

Par contre, j’ai pu me débrouiller tout seul pour installer win4lin (avec l’aide de l’assistance technique de l’éditeur du logiciel pour les imprimantes), j’ai en effet aussi dû installer mon vieux W98 avec win4lin (en version 3 que j’ai mis à jour en version 4) car je dois continuer d’utiliser certains logiciels, de plus en plus rarement, avec W98. Le logiciel win4lin (propriétaire) me donne entière satisfaction : il démarre W98 en moins de 15 secondes et les logiciels sont alors plutôt plus rapides et s’ils “plantent” : eh bien Linux est toujours là ! J’ai aussi installé plusieurs logiciels sans problème, mais certaines installations restent problématiques, c’est un point encore imparfait mais en rapide amélioration. On y arrive presque toujours mais parfois non sans peine. Mais désinstaller, rien de plus facile : on efface le répertoire et c’est fait (à moins de n’utiliser le gestionnaire de paquetages s’il a été installé avec lui).

Je ne dirais pas toutefois que je n’ai eu aucun problème. Ainsi cet été j’ai dû tout réinstaller à la suite d’une mauvaise tentative de reformatage d’une partition (pour diminuer l’espace consacré à Windows). Je ne sais si c’était une erreur de ma part ou un bogue de l’utilitaire de Mandrake, mais peut être ai-je couru après ? J’ai eu bien des problèmes avec Ghost de Norton qui ne respecte pas le mode de chargement de Linux, lilo en particulier ... Bref des problèmes j’en ai eus, mais ils ont été résolus et j’ai tendance à essayer de nouveaux logiciels, je suis donc un utilisateur à risque. Mais si KDE m’a quelques fois posé des problèmes (même un clavier figé), si Konqueror saute (rarement) à la lecture de certains sites, si OpenOffice s’est évanoui à la lecture de certains fichiers .doc, et si mon fils de 12 ans a réussi à planter le serveur X en essayant des jeux mal programmés (programmés pour une autre carte graphique que celle de mon portable) je n’ai jamais eu à redémarrer Linux (on tue au besoin le processus responsable du problème et on repart le serveur graphique), du moins à partir du moment où j’ai su le faire !

Je peux aujourd’hui confortablement et agréablement faire presque tout ce que je veux avec des logiciels libres, mais je dois avouer que j’ai eu besoin d’aide, probablement pas plus que bien de mes collègues qui utilisent Windows et les logiciels propriétaires choisis (?) par mon institution. Ceci a été rendu possible par l’étonnante progression de ces logiciels, et l’aide d’amis, d’assistants (cet été) et surtout du « réseau ». Ainsi quand j’ai dû installer le modem de ma machine avant d’aller donner quelques conférences, j’ai téléchargé le « winmodem » du site de IBM. Hélas ! j’ai échoué dans dans ma première tentative d’installation. J’ai alors (ma carte réseau fonctionnait) navigué et trouvé quelques explications, qui se sont révélées insuffisantes, mais j’avais ainsi dégrossi mon problème et localisé le forum des développeurs du type de logiciel qui me posait problème. Après quelques échanges l’un d’eux m’a demandé de lancer une commande et de lui envoyer le résultat ; il m’a alors répondu en me communiquant deux lignes à rajouter dans un fichier de configuration ; un autre m’a conseillé d’utiliser kpp au lieu du module que j’utilisais : et le modem fonctionnait. Ce jour là j’ai pris conscience, une fois de plus, de la communauté du logiciel libre mais aussi

de l'énorme base de connaissance qui a progressivement été constituée dans les divers serveurs connectés à Internet.

En conclusion, je n'ai pas encore pu totalement m'affranchir de certains logiciels propriétaires, soit parce que leur équivalent libre n'existe pas (Silverrun, par exemple), soit par manque de compatibilité des documents (MS-Office : mais la version OpenOffice 1.0 fait vraiment du bon travail), soit pour me rendre compatible (win4lin, par exemple). Mais est-ce vraiment là l'objectif ? Je ne vois pas de problème à utiliser certains logiciels propriétaires, quand il n'y a pas d'autre choix, mais je vois aussi bien des raisons à privilégier les logiciels libres quand ils existent. Par contre je peux maintenant, à la maison, éviter de recourir à des logiciels illégalement copiés : "Le Gimp" fait aussi bien que Photoshop (trop cher pour mon usage personnel, et faut-il le dire conseillé, voire prêté par des copains ?), je n'ai plus besoin de « recopier » à la maison MS-Office, je peux composer un document html facilement (et de bonne qualité), produire un document pdf, et bien d'autres choses. Je suis maintenant à l'aise avec Linux et KDE que je préfère aux autres bureaux que j'ai essayés. Je trouve maintenant que Word n'est pas un si bon outil que ça pour moi, il ne fait finalement qu'automatiser une machine à écrire ; j'aime bien mieux composer mes documents à l'aide de Lyx, c'est vraiment un bel outil fait pour créer des contenus pour les imprimer avec une belle qualité ; pour un petit document Kword est remarquable de puissance et de simplicité, un beau logiciel bien dessiné, et depuis que j'ai remarqué que l'utilitaire d'impression de KDE, kprinter, imprime les documents - stockés en format XML- directement sous forme de fichier pdf, j'ai une belle solution pour la création des petits documents. Mon entêtement à devenir un utilisateur des logiciels libres commence à trouver sa récompense : mon poste de travail est agréable et totalement fonctionnel. Mon environnement fortement « Microsofté » ne s'en est même pas rendu compte (sauf si je le leur ai dit) : je communique avec mes collègues sans leur poser de problème et quand il y a problème (ça arrive encore de temps en temps) : je me débrouille, au besoin en démarrant MS-Office sous win4lin (ou Cross Over que je viens d'essayer et qui exécute très bien MS-Office), après tout j'ai payé les licences, alors pourquoi ne pas m'en servir en attendant d'être débarrassé de certains formats propriétaires ! Parfois, de plus en plus, devrais je dire, je leur demande d'être un peu plus respectueux de mes choix, en particulier en utilisant des formats publics et si possible libres. De plus en plus ils comprennent et acceptent, mais ... il y a là aussi des têtes très dures ...

Daniel Pascot, Professeur, Université Laval, Québec, automne 2002