

**UNIVERSITES DE
PARIS XI (Sceaux) – PARIS I (Sorbonne)
FACULTES DE DROIT**



Mémoire présenté par M. JULIEN LINSOLAS
En vue de l'obtention du DESS Droit du Numérique et des
Nouvelles Techniques

Sous la direction de M. le Doyen Pierre SIRINELLI.

Année universitaire 2001-2002

A mes parents,

"on ne connaît que les choses qu'on apprivoise"

Antoine de Saint-Exupéry,

le petit prince.

LE LOGICIEL LIBRE

PRESENTATION GENERALE.

Sommaire

INTRODUCTION	5
TITRE I : LE LOGICIEL LIBRE ENCADRE PAR LE CODE DE LA PROPRIETE INTELLECTUELLE.....	9
CHAPITRE I : EN TANT QUE CREATION DE L'ESPRIT.....	9
<i>Section 1 : Une réservation par assimilation</i>	<i>10</i>
<i>Section 2 : les dérives de la propriété littéraire et artistique.....</i>	<i>17</i>
CHAPITRE II : EN TANT QUE PRODUIT INDUSTRIEL.....	24
<i>Section 1 : Une exclusion de principe.....</i>	<i>24</i>
<i>Section 2 : La remise en cause du principe.....</i>	<i>30</i>
TITRE II : UNE PROTECTION AD HOC.....	36
CHAPITRE I : LE RECOURS AU CONTRAT.....	37
<i>Section 1 : Typologie des diverses licences.....</i>	<i>37</i>
<i>Section 2 : Leur appréhension par le droit commun des contrats.....</i>	<i>40</i>
CHAPITRE II : LA CATHEDRALE ET LE BAZAR : L'APPROPRIATION DE L'INFORMATION.....	47
<i>Section 1 : La cathédrale, l'idée du tout appropriable.....</i>	<i>48</i>
<i>Section 2 : Le bazar, la circulation du savoir.....</i>	<i>50</i>

INTRODUCTION

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi)

PS. Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).

Ce n'est peut être pas le mail le plus célèbre d'Internet, mais c'est en tous cas l'un des plus fondateurs. Par ce message, posté le 25 août 1991, Linus Torvald donnait naissance à LINUX !

A l'origine de l'informatique, logiciels et ordinateurs étaient indissociables. La société IBM offrait gratuitement ses programmes lorsque ceux-ci n'étaient pas directement implémentés. L'entreprise cliente possédait également les codes sources livrés en même temps que le logiciel sans frais supplémentaires. Chaque adaptation et difficulté surmontée sur le code par les utilisateurs étaient renvoyées à IBM afin d'améliorer la qualité de ses logiciels. La société, en position dominante sur le marché, incitait même au partage des codes. Les clients jouaient le rôle de beta-testeurs évitant à IBM de

recourir à des batteries de tests dispendieuses¹. Avec l'apparition de la concurrence, la symbiose se brisa et la qualité des logiciels, devenus propriétaires, s'en ressentit. Développés et vérifiés en interne, les programmes devenaient instables et sujets à de nombreux bogues. Pour E.Moglen² « ...un mode lamarckien, dans lequel les améliorations peuvent être effectuées partout, par tout le monde et héritées par tous les autres, aurait effacé le déficit, restaurant dans le monde de l'ordinateur personnel la stabilité et la robustesse des logiciels écrits dans un environnement quasi propriétaire de l'ère des supercalculateurs. Mais le business model de Microsoft écartait l'héritage lamarckien des améliorations de logiciels. ». Internet allait rendre le modèle viable. L'interconnexion globale facilite l'échange dans un but d'innovation. La rationalisation et la distribution des résultats de ce partage s'en trouvaient améliorées.

Grâce au réseau des réseaux, il est désormais beaucoup plus facile pour des personnes physiquement éloignées de développer un logiciel en commun. Cette collaboration globale, conséquence directe de la diffusion d'Internet, est à l'origine du développement massif de ce que l'on nomme les « logiciels libres ». L'esprit du logiciel libre est né avec le Request For Comment (RFC) de l'équipe ayant participé à la naissance d'Internet en 1969. Mais le mouvement fut relancé par M. Stallman, en 1985 avec la création de la Free Software Foundation, et de la méthode du copyleft par la licence GNU GPL³.

D'après l'AFUL (l'Association Francophone des Utilisateurs de Linux et des Logiciels libres) « est considéré comme libre le logiciel disponible sous forme de code source, c'est-à-dire un logiciel dont l'architecture interne est partagée et diffusée librement, toute personne pouvant participer à l'élaboration du produit en proposant ses propres améliorations. ».

Par ailleurs libre n'est pas gratuit. Contrairement à une croyance fortement répandue, notamment par la confusion née de la mauvaise traduction du terme anglais "free software"⁴, un logiciel libre peut faire l'objet d'une exploitation commerciale. Il s'intègre sans difficulté dans une logique économique, et comme le fait remarquer Jon Hall, directeur de Linux Software, « There 's definitely money to be made in Linux ».

¹ Stratégie marketing connu sous le nom de FUD

² E.Moglen, in l'anarchisme triomphant, 1999, <http://emoglen.law.columbia.edu>

³ Il s'agit de la General Public Licence, voir à ce sujet le travail réalisé par M.M.Clement-Fontaine in "La Licence GNU", mémoire Montpellier 1999

⁴ Free pouvant signifier à la fois libre et gratuit, mais pour ce qui concerne un logiciel gratuit le terme employé est freeware (voir infra)

La différence entre le logiciel libre et le logiciel commercial dit "classique" réside dans la diffusion ou non du code source.

L'une des raisons pour laquelle le logiciel libre s'est révélé être une alternative viable face aux logiciels propriétaires tient à la qualité de ses produits qui émerge de ce que E. Raymond appelle "la loi de linux"⁵ : avec suffisamment d'yeux tous les bugs disparaissent ! Le logiciel libre a acquis une crédibilité certaine face aux logiciels classiques et l'on trouve Linux aussi bien gérant la navette spatiale qu'une partie de l'infrastructure des locaux de Microsoft à Redmond !!!

Le logiciel libre est au centre d'enjeux économiques considérables. De plus en plus d'entreprises, mais aussi des administrations, s'orientent vers des solutions de type linux ou openBSD. L'attrait commercial de licences moins onéreuses et contraignantes est l'une des raisons de cet engouement. Les sociétés ont tendance à se détourner des applications standards au profit d'alternatives plus exotiques plus souples, stables et performantes. Voué à prendre des parts de marché plus conséquentes, des commissions se penchent sur le problème⁶. Cependant les réticences sont encore nombreuses tant il est vrai qu'il existe autour du logiciel libre un flou juridique. L'absence de recul ainsi que l'inexistence de contentieux rendent frileux des juristes qui ne peuvent se référer à une quelconque réglementation. Toujours est-il que le logiciel libre représente une solution d'avenir et qu'à ce titre il convient d'en apprécier la teneur juridique.

Toute création de l'homme, un tant soit peu novatrice, est protégeable. L'instauration d'un droit privatif permet de prohiber tout acte non autorisé par l'auteur. Le logiciel libre ne fait pas exception, parfois au prix de quelques remaniements des régimes auxquels il se rattache. Mais par un renversement des fondements, tout acte est en l'espèce autorisé à l'exception de ceux expressément interdits. L'auteur accepte ab initio toute reproduction, représentation, distribution ou modification de son œuvre sans véritable contrepartie. Cela ne signifie pas qu'il y ait eu intention libérale de sa part. Rendre une œuvre libre n'est pas la faire tomber dans le domaine public. Elle reste pleinement protégée. Le logiciel libre est-il donc suffisamment spécifique, au point que le droit commun ne parvienne plus à l'appréhender, pour bénéficier d'un régime propre ? L'introduction de libertés dans un milieu d'appropriation et de réservation n'est pas soulever quelques problèmes.

⁵ Given enough eyeballs, all bugs are shallow." I dub this: ``Linus's Law"

⁶ Comme la commission Syntec réunissant Microsoft, Mandrake et Steria ou encore les réflexions de sociétés telles que Cap Gemini ou Altran.

C'est pourquoi se pose la question de savoir qu'elles peuvent être les nouvelles conditions de cette protection ? Cette liberté ne peut pas être omnipotente et il convient alors d'en apprécier l'étendue et le champ.

Ainsi elle est encadrée par le code de la propriété intellectuelle (Titre I) mais également par un régime ad hoc plus approprié (Titre II).

TITRE I : LE LOGICIEL LIBRE ENCADRE PAR LE CODE DE LA PROPRIÉTÉ INTELLECTUELLE.

Objet de droit insolite dans l'univers des créations numériques, le logiciel libre n'échappe pas à l'emprise du droit de la propriété intellectuelle. Comme toutes les œuvres atypiques, il a été question de savoir quel régime pouvait lui être appliqué.

Pour qu'une œuvre, ou une invention donne prise à une protection de la propriété intellectuelle, il ne lui suffit pas seulement d'exister et d'émaner de l'homme, encore faut-il qu'elle remplisse les conditions posées par la loi. Le problème réside dans la manière dont le droit de la propriété intellectuelle appréhende le logiciel libre. Et de ce choix dépend son avenir. Est-ce une invention, soumise au droit des brevets et de la propriété intellectuelle ? Ou, s'agit-il d'une œuvre obéissant alors à la propriété littéraire et artistique ? Opter pour cette dernière revient simplement à adapter le droit actuel. Proche de l'invention, sa protection deviendrait incertaine, tant les réticences à son égard sont fortes (Chapitre 2). A tel point que nombreux sont ceux qui estiment qu'un tel choix condamnerait le logiciel libre. Toutefois le classer en tant que tel est par trop réducteur et donc inexact. Le régime du droit d'auteur spécial lui conviendrait mieux, mais là encore ce rattachement doit se faire au prix de quelques ajustements théoriques (Chapitre 1).

CHAPITRE I : En tant que création de l'esprit.

Le logiciel est l'un des fruits de la révolution numérique. Le considérer comme une œuvre littéraire (section 1) n'allait pas sans poser quelques complications. Le régime de droit commun avait été instauré pour des œuvres dites classiques. Un premier ajustement est apparu alors indispensable, en donnant naissance au droit d'auteur spécial des logiciels. Devait-on franchir une seconde étape avec l'apparition du logiciel libre (section 2) ?

Section 1 : Une réservation par assimilation

Le choix d'un droit privatif pour le logiciel est très vite apparu comme une nécessité. En effet, l'appréhender par un mécanisme de propriété intellectuelle semblait être la solution la plus efficace pour lutter contre les pratiques de piratage qui sévissent dans un secteur où l'investissement est fort et le parasitisme facile. Pour autant déterminer lequel de ces mécanismes devait être retenu fut sujet à de vives controverses.

De part son caractère ambivalent, le logiciel n'entrait dans aucune catégorie préexistante. Proche du brevet par sa fonction technique, il est aussi une œuvre du langage (I) car, selon les termes du Professeur Lucas « *...sa création emprunte par nature le vêtement des mots.* » Une première approche par un droit Sui Generis fut alors envisagée. Mal accueillis par les professionnels, les divers projets furent abandonnés⁷. Finalement, le législateur opta pour une protection par le régime du droit d'auteur (II).

I) Le logiciel, une œuvre littéraire.

Or, le logiciel libre, avant d'être libre, est un logiciel. A ce titre son régime se fonde dans celui du logiciel alors même que son existence découle d'une logique différente. Voulu comme une sorte de négationnisme du copyright, le logiciel libre donne prise néanmoins au droit d'auteur. Il en résulte une réservation qui paraît contre nature (A) que tente d'apprécier le droit de la propriété intellectuelle (B).

A) Le choix d'une réservation contre nature.

Rapidement fermée (loi du 2 janvier 1968), la voie du brevet n'apparaissait plus comme une solution envisageable. Le recours aux dispositions de la loi de 1957 permettait alors de bénéficier d'une action en contrefaçon utile à la défense des intérêts des investisseurs. Mis en porte à faux, présenté comme une solution de rechange, ce choix reçut l'approbation des premiers juges qui eurent à statuer en la matière. Une fois

⁷ L'OMPI en 1983, celui de l'INPI du 15 décembre 1983....

ouverte, la jurisprudence s'engouffra dans cette voie, confortée par un accueil plutôt favorable de la doctrine. Ainsi le TGI de Bobigny⁸, le 11 décembre 1978, comme le Tribunal de Commerce de Paris⁹ rendaient un jugement en ce sens avant que la Cour d'Appel de Paris¹⁰, dans un arrêt du 2 novembre 1982, ne l'entérine. Mais c'est surtout par 3 arrêts célèbres¹¹ (affaires Pachot, Atari et Williams Electronics) que la Cour de Cassation scellait le sort de la protection du logiciel.

Consacrée par la loi Lang du 3 juillet 1985, cette solution s'articule autour de trois postulats¹². Tout d'abord réduire l'incertitude juridique résultant de possibles solutions jurisprudentielles contradictoires. Ne pas recourir ensuite à un droit *Sui generis*. Et enfin investir les logiciels de prérogatives connexes au droit d'auteur. La loi du 10 mai 1994, transposant la directive du 14 mai 1991, achève le travail législatif, qui s'avère, d'un point de vue international, être la norme¹³.

B) L'objet de la protection.

1) Le caractère atypique du logiciel libre.

Le logiciel se définit comme étant « *l'ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de l'information.* »¹⁴. Au-delà du simple code, la législation étend la notion de logiciel comme regroupant également la documentation auxiliaire. La jurisprudence en a même conclu, dans un arrêt de la Cour d'Appel de Versailles¹⁵ du 15 juin 1992, que ces deux éléments étaient indissociables. Il faut entendre par documentation, les éléments accessoires nécessaires au fonctionnement du logiciel et non les divers manuels qui restent soumis au droit commun de la propriété littéraire. Pour un logiciel libre se pose donc la question de savoir si doit être incluse dans cette analyse la documentation annexe ? Si celle-ci n'est pas directement relative au code source alors nous estimons qu'elle doit suivre le régime classique. Dans le cas contraire,

⁸ TGI Bobigny, 11 décembre 1978. Expertise 1982, n°39 p73

⁹ TC Paris, 18 novembre 1980. Expertises 1982, n°39, p74

¹⁰ CA Paris, 2 novembre 1982, Gazette du Palais 1983 p117

¹¹ Cass Plén, 7 mars 1986

¹² Rapport JOLIBOIS, in rapport sénatorial, Ch, t1 p55

¹³ Cf art 10 Convention Berne, ADPIC

¹⁴ Arrêté du 22 décembre 1981 sur l'enrichissement du vocabulaire de l'informatique, JONC 17.01.82

¹⁵ CA Versailles, 14^{ème} Ch, 15 juin 1992, n°877/92, in Lamy Droit de l'informatique 2002, n°158

deux hypothèses s'ouvrent à nous. D'une part la documentation relève plus de l'œuvre littéraire auquel cas une forme de modèle libre pourra lui être appliquée¹⁶ sinon elle devra se soumettre en tant que complément au code à la licence libre d'autre part.

2) Un logiciel polymorphe.

En droit, la notion de logiciel recoupe divers types d'objets. Il en est de même pour le logiciel libre. Certes, le plus connu d'entre eux, Linux, à tendance à occulter l'immense variété de formes que peut revêtir le logiciel libre, mais cela concerne tant les logiciels d'exploitation que d'application, les langages de programmation que certains algorithmes.

Cependant, le droit d'auteur protège indifféremment les logiciels sans qu'il soit besoin de les distinguer entre eux selon leur nature (cf l'affaire Apple TGI Paris, 21 septembre 1983). Quant aux langages de programmation, la problématique reste identique et le caractère libre du langage, tel Pearl, n'y change rien. En revanche, pour ce qui concerne les algorithmes, la réponse paraît moins évidente depuis l'adoption du principe libertaire par certains d'entre eux comme le DivX. ! De facto ce dernier devrait être analysé comme étant un logiciel, puisqu'un logiciel libre est avant tout un ...logiciel. Mais cette tautologie ne résout donc le problème que par une explication de fait et non de droit !

II) Recours au régime de droit commun.

En son article L 112-2 ¹³¹⁷, le CPI considère le logiciel comme une œuvre de l'esprit. Par conséquent il devient protégeable du seul fait de sa création (L 111-1)¹⁸ pourvu qu'il soit original (L 112-4)¹⁹, c'est-à-dire pour autant qu'il porte la marque de l'apport intellectuel de son auteur.

Avant l'ouverture du code à la communauté, le logiciel suit le régime de droit commun sans que cela ne soulève de problématique particulière. Mais qu'en est-il une fois le code ouvert ? Comment doit être qualifiée l'œuvre que de nombreuses personnes ont

¹⁶ Cf infra, Titre II, Chap II section 2

¹⁷ **Art. L. 112-2.** *Sont considérés notamment comme œuvres de l'esprit au sens du présent code... 13° Les logiciels, y compris le matériel de conception préparatoire...*

¹⁸ **Art. L. 111-1.** *L'auteur d'une œuvre de l'esprit jouit sur cette œuvre, du seul fait de sa création, d'un droit de propriété incorporelle exclusif et opposable à tous*

¹⁹ **Art. L. 112-4.** *Le titre d'une œuvre de l'esprit, dès lors qu'il présente un caractère original, est protégé comme l'œuvre elle-même.*

modifiée ? A qui appartiennent les droits d'auteurs résultant de ces modifications ? L'ajout de fonctionnalités ou la modification de l'œuvre originelle, par l'entremise des utilisateurs, peuvent-ils constituer les éléments d'une nouvelle œuvre, ou ne sont-ils que des apports supplémentaires à l'œuvre primaire, considérée comme étant en perpétuelle gestation ? (A) Et quant est il des prérogatives des auteurs ? (B)

A) Une genèse perpétuelle ?

1) Une œuvre inachevée ou dérivée ?

Constamment remanié, amélioré ou modifié, le logiciel libre semble osciller entre l'œuvre inachevée et l'œuvre dérivée. Cette conception d'œuvre en gestation n'est pas révélatrice de l'expérience empirique. Dans le milieu informatique, tout logiciel connaît son lot de patchs ou de plug-ins, sans pour autant être considéré comme une œuvre inachevée. Sachant qu'une œuvre est réputée créée du seul fait de sa réalisation, même inachevée (art L 111-2)²⁰, parler de gestation ne présente que peu d'intérêt. En revanche se pose ici la question de savoir quel est le processus de création de ce type de logiciel ?

A l'origine, un logiciel libre peut être le fruit de la création d'une seule personne. En l'absence de l'incorporation d'une œuvre différenciée et préexistante, l'œuvre créée est dite primaire et son auteur exerce sur celle-ci la plénitude de ses droits. Il s'agissait au début de l'ère informatique d'un processus courant. Aujourd'hui il est de plus en plus rare et ne permet pas d'aborder l'une des spécificités de notre sujet. En effet, le principal avantage du logiciel libre, et qui en constitue l'essence, est l'ouverture à une communauté de son code source. Véritable incitation à la création dérivée, il est par là même le plus souvent une œuvre seconde ou composite. C'est, selon la juste expression de Mme Clément-Fontaine²¹, ce mouvement que tente d'appriivoiser le droit. La mise à disposition aux utilisateurs du code, mais également des éléments nécessaires à la conception, favorise la modification intrinsèque du logiciel. Ces transformations consistent, notamment en l'ajout de fonctionnalités, en l'optimisation du moteur ou par l'adjonction d'un autre programme. Une nouvelle

²⁰ **Art. L. 111-2.** *L'œuvre est réputée créée, indépendamment de toute divulgation publique, du seul fait de la réalisation, même inachevée, de la conception de l'auteur.*

²¹ Précitée.

version du logiciel pourra en découler, ce qui va au-delà de la simple adaptation prévue par le CPI à l'article L 122-6-1.

Selon l'article L 113-2, une œuvre dérivée correspond à « *...l'oeuvre nouvelle à laquelle est incorporée une oeuvre préexistante sans la collaboration de l'auteur de cette dernière.* ». Protégée en tant que telle, sans préjudice des droits de l'auteur de l'œuvre originale (L 112-3), l'œuvre seconde jouit d'une certaine autonomie. Dès lors, à la condition d'avoir obtenu l'autorisation, l'auteur de l'œuvre composite garde en principe tout contrôle sur la destination de son œuvre. Or, d'après les pratiques courantes existantes dans le milieu du libre, l'œuvre dérivée tombe par elle-même dans cette communauté. Juridiquement, il est permis de douter de la validité d'une telle obligation contractuelle²².

Par ailleurs, et sans que cela soit exclusif de la qualification d'œuvre dérivée, le logiciel libre peut être conçu par plusieurs personnes. S'agit il alors d'une œuvre collective ou de collaboration ? La première se définit comme « *...étant l'oeuvre créée sur l'initiative d'une personne physique ou morale qui l'édite, la publie et la divulgue sous sa direction et son nom et dans laquelle la contribution personnelle des divers auteurs participant à son élaboration se fond dans l'ensemble en vue duquel elle est conçue, sans qu'il soit possible d'attribuer à chacun d'eux un droit distinct sur l'ensemble réalisé* ». Dans cette hypothèse, la problématique envisagée dans le paragraphe précédent tombe d'elle-même. Investie ab initio des droits sur l'œuvre, la personne qui en prend l'initiative et la publie sous son nom n'a plus à faire la preuve de sa qualité de cessionnaire. Chaque contribution des internautes tombe dans l'escarcelle du coordinateur et une seule personne bénéficie de l'intégralité des droits. Confrontée à la réalité, cette hypothèse ne tient plus, à moins que le logiciel ait été conçu au préalable en interne (au sein d'une entreprise ou d'un organisme spécialisé (AFUL, FSF...)). En effet, il est difficilement concevable d'admettre que les différents intervenants sur un logiciel, et cela d'autant plus si celui-ci a vu son code source divulgué sur Internet, reçoivent d'une personne unique des directives et instructions, éléments pourtant nécessaire à la qualification en œuvre collective.

Dans la mesure où chaque internaute ayant participé à l'élaboration du logiciel a fourni un effort intellectuel dissocié, il paraît plus juste d'envisager une qualification en œuvre de collaboration. Mais au-delà d'une pluralité d'apports, il faut qu'il existe un travail créatif concerté et conduit en commun. A se contenter de la définition légale de

²² Cf Développement dans le 1^{er} Chap du Titre II.

l'œuvre de collaboration²³, un logiciel libre peut être défini comme tel. L'œuvre n'est que l'agrégat de contributions diverses. A cela, il faut apporter une double nuance. D'une part, toutes les contributions n'ont pas la même valeur. Pour que l'auteur de l'une d'entre elles soit reconnu comme l'auteur de l'œuvre globale encore faut-il que son apport soit réalisé avec une vision générale de l'œuvre finale. D'autre part, l'œuvre de collaboration n'existe qu'en l'absence d'autre qualification (hormis celle composite).

Dès lors qu'un logiciel est modifié de façon substantielle, en résulte-t-il une œuvre dérivée ? Un logiciel est une œuvre complexe composée de divers éléments distincts. De par son caractère "viral"²⁴, en ce sens qu'un logiciel libre soumet tout code dérivé aux termes et conditions de l'œuvre primaire, l'ensemble du logiciel devient libre. Il s'agit là d'une sorte d'accession à l'œuvre principale. Paradoxalement, devenu libre, cet élément n'est pas inéluctablement intégré dans l'œuvre primaire et en constituer une nouvelle version. Cela peut être un patch ou un pilote. Pourtant d'un point de vue juridique, leur réunion doit être considérée comme donnant naissance à une œuvre composite.

2) Une forme originale.

Condition préalable à une protection par le droit d'auteur, le logiciel doit présenter un caractère original. Remaniée et adaptée, la notion d'originalité relève d'une approche éminemment objective. Nouveauté dans l'univers des formes pour M. Vivant²⁵ ou effort intellectuel individualisé pour M. Le Stanc²⁶, l'originalité s'entend désormais comme la marque de l'apport intellectuel²⁷. Pour le logiciel libre, l'originalité doit apparaître au niveau du code source primaire. Les différents apports ultérieurs n'ont pas nécessairement à être originaux. Ceux qui le sont, s'incorporent irrémédiablement dans la nouvelle œuvre dérivée et l'originalité s'apprécie de la même manière que pour toute autre œuvre composite.

B) Des prérogatives diminuées.

²³ **Art. L. 113-2.** Est dite de collaboration l'oeuvre à la création de laquelle ont concouru plusieurs personnes physiques

²⁴ Cf Chap 1 Titre 2

²⁵ M.Vivant, in brèves réflexions sur le droit d'auteur, revue informatica diritto, 1984

²⁶ M.Le Stanc, in logiciel, originalité et activité créative dans la loi de 1985, Expertises, 1986, n°81, p32

²⁷ Arrêt Pachot, Ass Plen, 7 mars 1986.

1) Un abandon des droits patrimoniaux.

En offrant à la communauté le code source de son logiciel, l'auteur n'exprime pas la volonté de mettre l'œuvre dans le domaine public, mais renonce uniquement à la plupart de ses droits patrimoniaux. En d'autres termes, il organise sa mise à disposition. Il n'y a pas d'abdication, ni transfert de droit de propriété. L'auteur n'organise pas la cession de ses droits. Non seulement cette session ne se présume pas mais encore le CPI impose-t-il un formalisme rigoureux qui ne se retrouve pas en l'espèce. Ay regarder de plus près, l'auteur qui sacrifie son œuvre sur l'autel du libre se dessaisit de fait de l'ensemble de ses droits patrimoniaux. En outre, il n'est pas de l'essence du copyleft d'accorder des droits exclusifs. Dès lors, l'auteur, tout du moins la personne qui accorde la licence, conserve ses droits afin de contrôler au mieux toute évolution du logiciel. Mais ces droits, l'auteur ne les exerce pas librement ! Bien au contraire, les usages²⁸ de la communauté mettent par surcroît à sa charge un certain nombre d'obligations contraignantes. La principale lui incombant est celle d'une mise en jouissance, c'est-à-dire qu'il doit aménager la libre utilisation, diffusion et modification de l'œuvre. Pour cela, il abandonne nécessairement une part non négligeable de ses prérogatives. Ainsi le droit de reproduction, devenu absolu avec la disparition du droit à la copie privée (art L 122-6-1) se réduit en l'espèce au minimum. Il ne disparaît pas. Il est suspendu par la destination du logiciel. Le bénéficiaire a la faculté d'effectuer autant de copie qu'il le désire.

Pour Mme M.Clement Fontaine²⁹, il ne s'agit pas d'un abandon mais du simple usage de ses droits : «...*le donneur de licence use de son droit de son droit patrimonial en offrant au licencié la liberté de diffuser le logiciel...* ». Cette liberté peut résulter de la volonté de l'auteur. Il lui appartient d'interdire ou d'autoriser tout acte sur son oeuvre, en vertu de son droit exclusif. Cependant, ce choix ne lui est pas toujours permis. Le caractère viral du logiciel libre infecte toute œuvre dérivée. L'auteur de cette dernière a pour obligation de garantir des libertés similaires aux tiers auxquels le logiciel modifié est redistribué. C'est pourquoi, il est préférable de parler d'abandon plutôt que d'usage.

Quant aux droits de représentation, la problématique reste inchangée. L'œuvre protégée se distingue des effets audiovisuels produits par le logiciel. En ce cas, il n'y a

²⁸ Cf Titre 2 chap 1

²⁹ Mémoire DEA précité

représentation que lorsque le code source est communiqué au public. Autrement dit, un logiciel ne peut être libre que si l'auteur, là encore, abandonne ses droits.

2) La persistance des droits moraux.

Coeteris paribus, l'auteur n'abandonne en aucun cas ses droits moraux ! Attachés à la personne même de l'auteur, ces droits sont perpétuels, inaliénables et imprescriptibles (L 121-1)³⁰.

En matière de logiciel ils sont limités, pratiquement inexistant, puisque n'a été retenu par le législateur que le minimum, c'est-à-dire le droit à la paternité et celui de divulgation. Le droit de retrait disparaît comme celui du respect à l'intégrité de l'œuvre (L 121-7)³¹. Dans une économie de l'informatique dominée par le système de copyright américain, cette relative absence du droit moral n'est pas un obstacle, bien au contraire. Ainsi, l'auteur d'un logiciel libre ou de l'une de ses composantes, peut se prévaloir uniquement d'un droit à la reconnaissance !

Section 2 : les dérives de la propriété littéraire et artistique.

Dire qu'il existe des dérives est exagéré. Simplement le droit de la propriété intellectuelle a quelques gênes à déterminer l'objet en question (I) et souffre d'une certaine inadaptation (II)

I) La difficile détermination de l'objet.

Sans cesse remanié, le logiciel libre est une œuvre qui évolue. Mais cette évolution s'effectue de manière relativement anarchique, de sorte qu'il devient difficile

³⁰ **Art. L. 121-1.** *L'auteur jouit du droit au respect de son nom, de sa qualité et de son oeuvre. Ce droit est attaché à sa personne. Il est perpétuel, inaliénable et imprescriptible. Il est transmissible à cause de mort aux héritiers de l'auteur.*

³¹ **Art. L. 121-7.** *Sauf stipulation contraire plus favorable à l'auteur d'un logiciel, celui-ci ne peut :*
 1° *S'opposer à la modification du logiciel par le cessionnaire des droits mentionnés au 2° de l'article L. 122-6, lorsqu'elle n'est préjudiciable ni à son honneur, ni à sa réputation ;*
 2° *Exercer son droit de repentir ou de retrait.*

d'en appréhender l'objet mis à disposition (B). C'est pourquoi, toute évolution substantielle doit se démarquer de la simple idée (A).

A) Distinction entre idée et expression originale.

1) Une exclusion de principe.

Il est de notoriété juridique que l'idée n'est pas un objet de droit. D'une part parce qu'elle relève du fonds commun qui est de libre parcours et que d'autre part elle requiert une certaine forme d'expression pour être protégée. Pour autant ce n'est pas la condition de mise en forme qui doit être avancée mais plutôt celles de la précision³² et de l'originalité. Bien que la notion de forme doive être analysée à l'aune de la précision de son contenu. Ce qui est véritablement exclu est la pensée, à peine esquissée, les règles abstraites et les principes généraux. Ainsi un algorithme tient de l'idée et de la méthode s'il n'apparaît pas sous une forme exprimée et à la condition en outre que celle-ci soit originale. Un algorithme consiste en une méthode mathématique, une sorte de cheminement intellectuel. La rédaction de cette méthode s'effectue en dehors de tout processus de programmation. Il n'est en effet pas dépendant d'un quelconque langage de programmation. Selon M. Agbeshie³³, « *une méthode se définit comme la marche rationnelle de l'esprit pour arriver à la connaissance ou à la démonstration d'une vérité* ». Derrière toute méthode se cachent alors idées, concepts ou principes appartenant au patrimoine intellectuel de l'humanité, et qui ne sont pas susceptibles d'une quelconque appropriation. Cela relève du monde des idées et entre donc de ce fait dans le domaine public. Néanmoins, la méthode algorithmique organise, de manière logique, ces éléments. Dès lors est-il possible de s'approprier cette structuration sous la forme écrite d'un algorithme ? L'insertion d'un codec³⁴ dans un logiciel libre ne devrait entraîner aucune conséquence du point de vue des droits d'auteur. Dans la pratique, un tel logiciel évolue le plus souvent par des ajouts minimes, de simples corrections plus proches de l'idée que de la véritable création informatique. Cependant, dès lors qu'un code existe, il devient difficile pour un juge de nier l'existence d'une œuvre logicielle.

³² M.Gauthier in « propriété littéraire et artistique », 4^{ème} édition, N°31

³³ Georges Ako. AGBESHIE, Protection des Formats de Compression Numérique, Mémoire DESS droit du numérique 1999.

³⁴ Codec = Terme désignant un algorithme de compression-décompression.

Par conséquent, la multitude d'apports rend incertain le respect des droits tant sur l'œuvre au ceux d'autrui.

2) Une protection indirecte.

Le risque fondamental de l'élargissement du champ du protégeable consiste en la potentialité accrue d'une contrefaçon sur l'un des éléments inclus dans le logiciel libre. Ces derniers, dans la plupart des cas, sont peu complexes et devraient donc être analysés comme étant de simples méthodes exclues de la sphère privative³⁵. Or, celle-ci tend à s'agrandir sous l'influence et la pression de certains lobbies. L'affaire de la corbeille du système d'exploitation MacOS illustre parfaitement la difficulté à dissocier l'idée de son expression matérielle originale. Est-elle protégée en tant qu'œuvre graphique, auquel cas ses fonctionnalités restent hors du champ de la protection, ou bien cherche t'on à protéger principalement la fonction ?

Le droit d'auteur en abaissant le seuil d'originalité requis permet d'accéder à une protection détournée des idées et principes. Cela s'exprime notamment par l'interdiction faite à l'utilisateur de décompiler le logiciel, opération pourtant essentielle à l'obtention des méthodes qui le sous-tendent. C'est la raison pour laquelle, en permettant un accès libre au code, le logiciel libre échappe à cette dérive du droit commun.

B) La mise à disposition.

1) Ce qui est remis.

Par l'article 1615 du code civil³⁶, la délivrance de la chose, en l'occurrence le logiciel, doit s'accompagner des accessoires (le code source), qui sont nécessaires à sa jouissance. La jurisprudence a considéré que le code source n'entraîne pas dans la définition des accessoires³⁷ puisque l'utilisateur dispose d'une version en langage évolué apte à répondre à ses besoins. Le code source n'est alors pas indispensable. Par ailleurs, aucune obligation légale née du CPI n'exige une telle remise. C'est donc à une logique totalement différente que répond le copyleft. Cette remise est de l'essence même des

³⁵ Cf supra.

³⁶ **Article 1615.** *L'obligation de délivrer la chose comprend ses accessoires et tout ce qui a été destiné à son usage perpétuel.*

³⁷ CA Paris, 11 janvier 1984, jurisdata n°20003

diverses licences. Aussi en imposant la délivrance, non seulement du logiciel mais également des sources, cette mise en jouissance spécifique au logiciel libre, « *tend à faire disparaître toutes les entraves au libre usage du logiciel organisées par le droit d'auteur* »³⁸. Mais quid des annotations, explications et documentations relatives au code ?

2) L'intelligence partagée.

L'existence et la sauvegarde des intérêts de l'auteur repose sur un régime fondé sur la propriété. Le mouvement du libre s'appuie sur des bases différentes. D'aucuns y ont même vu, comme R. Barbrook³⁹, la naissance d'un "cybercommunisme"! Sans atteindre de tels extrêmes, nous préférons parler d'intelligence partagée, consistant en une mise en commun des informations, idées et moyens sans pour autant exclure a priori toute notion de propriété. Selon Platon, l'idée est unique et statique. Avec les réseaux, elle devient multiple, mouvante et disponible à tous au même moment.

Or, comme nous l'avons vu précédemment, l'idée, lato sensu, peut faire l'objet d'une réservation. Lorsque la propriété intellectuelle étend son champ et sort de ses frontières, elle représente un obstacle à la créativité. Cependant, et nous suivrons sur ce point l'analyse de M. P.Levy⁴⁰ « *...la propriété collective librement choisie [...] suppose l'existence préalable, disponible, garantie par la loi, de la propriété intellectuelle. Seule l'inscription dans le droit de la propriété privée permet l'exercice de la liberté et de la responsabilité individuelle et par exemple, le renoncement à la propriété intellectuelle.* »

II) L'inadaptation du droit.

A) La décompilation

1) Une exception en droit commun.

³⁸ Mélanie Clement Fontaine, in mémoire DEA précité.

³⁹ R.Barbrook, in « the californian ideology », disponible sur www.hrc.wmin.ac.uk

⁴⁰ P.Levy, in « l'anneau d'or », Multitudes, mai 2001

Admise à titre d'exception (art L 122-6-1)⁴¹ la décompilation permet la reproduction de l'œuvre à des fins d'interopérabilité. La directive européenne de 1991 décrit l'interopérabilité comme « *l'interconnexion et l'interaction fonctionnelle qui peut être définie comme étant la capacité d'échanges de données et d'utiliser les données qui ont été échangées* ». Cette exception vise à permettre à l'utilisateur de procéder aux modifications nécessaires à la complète compatibilité du logiciel avec les autres éléments du système. Toujours est-il que cette décompilation n'est autorisée que lorsqu'elle s'avère indispensable. Nonobstant, ce droit est restreint par le fait qu'il ne peut être invoqué que si les informations n'ont pas déjà été rendues publiques, s'il porte uniquement sur la partie du logiciel nécessaire à l'interopérabilité et s'il ne porte pas atteinte à l'exploitation du programme (article 6 de la directive européenne, codifié à l'article L122-6-I-IV du CPI).

Hormis cette exception légale, l'utilisateur, bien que légitime, ne peut avoir accès au code source. Dans un arrêt du 6 octobre 1994, la Cour d'Appel de Versailles a interprétée de manière restrictive cette disposition et en l'absence de clause contractuelle, l'auteur ne peut être contraint à livrer ou communiquer à un tiers ses codes. Il apparaît donc qu'en droit commun, la divulgation des codes n'est possible qu'en de rares hypothèses, strictement encadrées. Les industriels⁴², prônant d'énormes enjeux économiques sont réticents à tout élargissement. Ils arguent du fait que la décompilation n'est pas indispensable à l'interopérabilité car la compatibilité est une donnée factuelle et parfaitement intégrée par les éditeurs, mais également elle représente un frein au développement⁴³. Cette dernière analyse se trouve confortée par le possible

⁴¹ **Art. L. 122-6-1. I.** Les actes prévus aux 1° et 2° de l'article L. 122-6 ne sont pas soumis à l'autorisation de droit d'effectuer...

IV. ...est indispensable pour obtenir les informations nécessaires à l'interopérabilité d'un logiciel créé de façon indépendante avec d'autres logiciels, sous réserve que soient réunies les conditions suivantes :

1° Ces actes sont accomplis par la personne ayant le droit d'utiliser un exemplaire du logiciel ou pour son compte par une personne habilitée à cette fin ;

2° Les informations nécessaires à l'interopérabilité n'ont pas déjà été rendues facilement et rapidement accessibles aux personnes mentionnées au 1° ci-dessus ;

3° Et ces actes sont limités aux parties du logiciel d'origine nécessaires à cette interopérabilité. Les informations ainsi obtenues ne peuvent être :

1° Ni utilisées à des fins autres que la réalisation de l'interopérabilité du logiciel créé de façon indépendante ;

2° Ni communiquées à des tiers sauf si cela est nécessaire à l'interopérabilité du logiciel créé de façon indépendante ;

3° Ni utilisées pour la mise au point, la production ou la commercialisation d'un logiciel dont l'expression est substantiellement similaire ou pour tout autre acte portant atteinte au droit d'auteur.

⁴² Par exemple l'association SAGE (Software Active Groupe for Europa)

⁴³ Point de vue particulièrement avancé aux USA pour l'inverse ingenering où la constitution impose le progrès comme condition.

échec devant le test des trois étapes⁴⁴ prôné par la directive du 22 mai 2001. Faire acte de décompilation suppose la traduction du code objet dans son intégralité afin de déterminer quelles sont les parties nécessaires à l'interopérabilité. En somme, et malgré la technique de la "clean room", l'utilisateur accède à la totalité du code de manière tout à fait légale et sous couvert d'inverse engineering. Un concurrent peut bénéficier des secrets de programmation. Néanmoins cette analyse est erronée à plusieurs égards. Etant peu rentable la décompilation n'est pas véritablement une menace économique. Elle n'est pas non plus un frein puisqu'elle favorise la standardisation.

2) Un principe pour les logiciels libres.

Toutefois la décompilation n'effectue seulement qu'une opération visant à traduire le code objet en code d'assemblage sans pouvoir obtenir la véritable structure originelle du code source. Pour les logiciels libres, cette exception ne trouve pas matière à s'appliquer. Le principe fondamental de libre accès au code rend inutile la mise en œuvre d'une telle prérogative. La décompilation selon M. E. DI FILIPPO⁴⁵ se manifeste dans un ordre public de direction. C'est-à-dire qu'elle a pour but de libérer l'utilisateur de la tutelle de l'auteur. Or en l'espèce, l'utilisateur n'a plus vocation à être protégé. Renverser l'exception en principe tend à simplifier le régime du logiciel et résout par la même les difficultés d'application de la décompilation.

B) Un rejet du droit d'auteur.

Le logiciel libre est né d'un mouvement à contre courant de celui du copyright et du droit d'auteur. Il apparaît alors paradoxal d'offrir au logiciel libre une protection issue d'une logique privative. Par essence, le droit est une matière d'ordonnement et de classification dont les principaux outils sont la qualification et le syllogisme. Dès lors, par un analogisme sémantique le logiciel libre se voit conférer un régime contre nature. Est-ce à dire pour autant que la protection est inappropriée et le droit incapable de saisir le particularisme de ce type de logiciel ? A cela, il convient de répondre par la

⁴⁴ **ART 5.5** *Les exceptions et limitations prévues aux paragraphes 1, 2, 3 et 4 ne sont applicables que dans certains cas spéciaux qui ne portent pas atteinte à l'exploitation normale de l'oeuvre ou autre objet protégé ni ne causent un préjudice injustifié aux intérêts légitimes du titulaire du droit.*

⁴⁵ E. Di Filippo, In mémoire DEA Sophia Antipolis, 1999

négative. Car bien au contraire, la souplesse du droit d'auteur permet, tant bien que mal diront certains, d'étendre son emprise aux logiciels libres.

CHAPITRE II : En tant que produit industriel.

Un logiciel libre n'a rien en soi d'une invention à proprement parler. Alors à la question le logiciel est-il brevetable ? Le juriste doit reformuler la problématique pour l'orienter plutôt vers l'objet de cette réservation. La confusion entre le logiciel-œuvre et le logiciel-invention résulte d'une certaine assimilation par métonymie des effets du programme à son format. Dès lors il est plus pertinent de se poser la question de savoir si le droit de la propriété industrielle est adapté, au moment où divers projets législatifs vont dans ce sens. C'est pourquoi on s'attardera sur une exclusion encore d'actualité (section 1) avant d'analyser les raisons d'une remise en cause du principe (section 2).

Section 1 : Une exclusion de principe.

Il n'est pas dans l'intérêt de l'industrie du libre d'opter pour la brevetabilité. En ce sens l'argumentation prônant son rejet revêt toute son importance. C'est avec d'autant plus de vigueur qu'elle trouve en l'espèce matière à s'appliquer. C'est après avoir étudié les fondements du rejet de la brevetabilité (I) que l'on verra en quoi cela pourrait faire peser une menace sur la communauté du libre, si le droit venait à admettre le brevet de logiciel. (II)

I) Fondement du rejet.

Rapidement un consensus s'était dégagé autour d'une protection du logiciel par le droit d'auteur. Dès la loi du 2 janvier 1968, il était acquis que ne pouvaient être brevetés les programmes ou séries d'instructions. C'est donc sans surprise que la loi du 3 juillet 1985 entérinait cette position et accordait une protection au titre de la propriété intellectuelle. Ce choix, bien que fortement influencé par des considérations politico-économiques, résulte pour une grande part de la difficulté pour le logiciel de répondre aux conditions posées par le droit de la propriété industrielle. Ainsi l'appréciation du

caractère technique, condition sine qua non pour l'accès à la protection, n'est pas évidente et il en va de même pour la contribution à l'état de la technique.

A) Appréciation malaisée du caractère technique.

L'invention doit être de nature technique. Malgré un postulat simple, il n'en demeure pas moins que le législateur s'est gardé de le définir ce qui amène à quelques divergences jurisprudentielles. A l'instar de la notion d'originalité, cette malléabilité permet une adaptation à de nombreuses inventions. Mais pour qu'un logiciel soit considéré comme une invention encore faut-il qu'il soit une solution technique à un problème technique et susceptible de répétition. Or, si le problème est évident, l'existence d'une solution technique adéquate l'est moins.

1) L'existence d'un problème technique.

Selon la théorie du noyau essentiel, le caractère technique dépend du domaine dont relève son élément essentiel⁴⁶. En somme il ne convient d'analyser qu'une partie uniquement de l'invention, ce qui pour un logiciel revient à déterminer l'aspect technique de son code source. Dans une approche restrictive, où la caractéristique fonctionnelle d'une revendication doit être en rapport avec l'ensemble des autres caractéristiques techniques, un logiciel ne présente aucun problème technique. Cependant la jurisprudence en la matière a sensiblement évolué, notamment avec la décision Sohei⁴⁷, abaissant par là même le niveau d'exigence. Dès lors, la CRT a plutôt fait prévaloir comme critère "les considérations techniques" marquant ainsi une dilution du caractère technique⁴⁸. Désormais, ce n'est plus tant dans le code source qu'il se révèle, mais dans un stade préalable à toute programmation, plus proche de l'activité inventive. Ce glissement du technique vers l'activité inventive induit directement l'existence d'un caractère technique à l'ensemble de l'invention. L'examen de cette activité s'effectue selon la procédure dite de "problem-solution approach". Pour un logiciel la présentation d'informations et leur traitement constituent-ils alors la solution technique au problème revendiqué ?

2) L'absence d'une solution technique.

⁴⁶ CRT, 21 mai 1987, équipe radiologique, PIBD 1988, 432 III, p185

⁴⁷ T769/92, 3.5.1, Sohei/système de gestion universel, Jo OEB 1995, p34

⁴⁸ Cf MM Aubert et Macrez, in mémoire DEA Montpellier, 2001

Pour reprendre l'analyse de MM. Aubert et Macrez⁴⁹, le caractère technique d'un logiciel ne peut résulter des seules impulsions électriques. Cela impose nécessairement qu'un effet technique supplémentaire existe, qui ne peut provenir que de la fonction qu'il exerce dans son interaction avec la machine.

Une solution n'a une nature technique que lorsqu'elle produit un effet technique. Admettre cela pour un logiciel, c'est renier le travail de la doctrine qui s'est efforcée d'assimiler un programme à une œuvre. En effet, son exclusion expresse du brevet conclue à la négation d'une solution technique. Aussi, le fait que la fonction du logiciel ne se révèle qu'à partir du moment où il est exécuté sur un ordinateur ne lui confère pas par la même un effet technique. Les impulsions électriques créées, manifestations matérielles de cet effet, a certes pour conséquence d'apporter des modifications physiques à l'ordinateur mais celles-ci sont communes et présentes pour tout type d'instructions. En déduire le caractère technique du seul fait qu'il s'agisse d'un programme d'ordinateur amène à effectuer une appréciation selon le genre. La solution technique passe alors nécessairement par l'effet technique supplémentaire. Consacrée que récemment par la jurisprudence⁵⁰, elle n'apporte pourtant pas de réponse satisfaisante. En effet, et là encore nous rejoignons le point de vue de MM. Aubert et Macrez, cet effet technique « *...allant au-delà des interactions physiques normale* » entre le logiciel et l'ordinateur est produit par les périphériques. Ce critère n'a de "supplémentaire" qu'en ce qu'il existe un matériel supplémentaire, breveté en combinaison avec le logiciel qui le fait fonctionner !

Donc à moins de considérer la solution technique au regard des améliorations des ressources apportées par le logiciel (à l'instar de la notion d'utilité anglo-saxonne), nous ne pouvons conclure ou à son absence en la matière.

B) La contribution à l'état de la technique.

Toute invention doit apporter une contribution à l'état de la technique de manière non évidente pour l'homme du métier. Le logiciel doit être innovant.

1) Le caractère inventif.

⁴⁹ In mémoire DEA précité

⁵⁰ Dec T1173/97 du 1^{er} juillet 1998, Jo OEB 1999, p620

L'article L 611-14 du CPI dispose « *qu'une invention est considérée comme impliquant une activité inventive si, pour un homme du métier, elle ne découle pas d'une manière évidente de l'état de la technique...* ». Toutefois une grande majorité des logiciels ne rempliraient pas cette obligation. Comme nous l'avons vu, l'activité inventive apparaît aussi bien lors de la solution apportée qu'au stade du problème posé. Pour M. Mathely⁵¹ « *...c'est l'énoncé du problème qui n'est pas évident...* », et non sa connaissance, ce qui reviendrait à l'intégrer dans l'état de la technique, « *...et c'est dans cet énoncé que réside l'activité inventive.* » Dès lors, toute modification d'un logiciel libre due à la réécriture de cet énoncé pourrait être en soi brevetable. Ce n'est pas l'écriture du programme qui est révélatrice de cette activité, qui relève plutôt du savoir faire, mais l'analyse et l'appréciation du problème. Aussi le juge se doit d'être rigoureux et restrictif dans son approche afin de ne pas protéger le seul problème à la base du logiciel et donc indirectement les concepts et méthodes qui s'y affèrent.

2) Le caractère nouveau.

Une invention pour être brevetable ne doit pas se retrouver dans l'état de la technique. Mais l'existence d'antériorité remet elle en cause son caractère nouveau ? La pratique de l'INPI (art L 612-12 7° CPI) ne retient que le défaut manifeste de nouveauté. Un logiciel partiellement nouveau peut donc être breveté. Tout tient à l'absence d'un état de la technique déterminé et pertinent. Et cela se complique d'autant plus avec la multiplicité des apports, modifications et nouvelles versions propres aux logiciels libres. La recherche d'antériorité relèverait plus de la gageure que du bon sens. De plus l'antériorité doit être de toute pièce pour qu'elle puisse être admise. Cela pose donc un réel problème pour la brevetabilité des logiciels libres. L'invention doit se retrouver en elle-même, à l'identique⁵², ce qui est rarement le cas puisque, par le biais de l'accès au code source, le logiciel est aisément modifiable sans altération de sa substance. C'est pourquoi un brevet portant sur un logiciel libre présente un danger pour l'industrie informatique.

II) Un péril pesant sur la communauté du libre.

⁵¹ Mathely, in « le nouveau droit français des brevets d'invention », 1991, p95

⁵² F.Pollaud-Dullian, Droit de la propriété industrielle, Montchrestien, 1999, p206 n°479

Considéré comme une ineptie par les défenseurs du modèle libre, le brevet de logiciel représente une véritable menace pour les éditeurs et informaticiens tant du point de vue économique que juridique.

A) Une menace économique.

Il est dans la nature du brevet de promouvoir l'innovation⁵³. A la fois protection de l'investissement et incitation à la recherche, la propriété industrielle devient une arme économique dans un marché oligopolistique. Fortement restructuré, le marché de l'édition logicielle est tenu par peu de firmes. Selon ces principaux acteurs, l'innovation n'est possible à long terme que si elle est soutenue par des brevets. Les éditeurs de logiciels libres, tels Red Hat, Mandrake (OS sur noyau linux), TCX (SGBD MySQL)⁵⁴, ne mettent pas en avant leurs brevets pour obtenir des fonds de capital risque⁵⁵. A contrario, certains avancent (comme M. Breese) le fait qu'un système de brevets peut permettre aux petites sociétés de résister plus facilement face aux grandes. Cela leur confère une protection de leur innovation contre toute reprise par ces dernières. Mais c'est oublier le prix d'un brevet comme sa puissance contraignante. Cette arme n'est efficace que si l'entreprise possède une assise financière conséquente, capable d'absorber le coût du dépôt de nombreux brevets.

De moyen de défense, le brevet devient une arme offensive. Par cette pratique, les majors de l'industrie cherchent à contrôler le marché, ou pour le moins encadrer toute nouvelle mise sur celui-ci d'un logiciel qui pourrait comporter en son sein une pseudo invention. Les concepteurs de logiciels libres sont pour la plupart du temps des particuliers ou alors des sociétés qui font justement appel à eux pour optimiser leurs codes, comme par exemple Netscape, dont le code mozilla est détenu par AOL.

B) Une menace juridique.

1) La peur du tout brevetable.

⁵³ Cf le titre du livre vert de la commission européenne « promouvoir l'innovation par le brevet »

⁵⁴ Ils se sont d'ailleurs associés dans une campagne contre la brevetabilité des logiciels. Disponible sur le site www.freepatents.com

⁵⁵ Cf Mémoire de M.Di Filippo, précité

Retenir un mode de protection dont les critères d'accessibilité restent flous et incertains pose le problème de la délimitation de son champ. C'est pourquoi d'aucun se montre réservé quant à l'intérêt d'une brevetabilité du logiciel. Le manque de rigueur dans le contrôle effectué par les divers organismes agréés, risque d'avoir pour conséquence le dépôt de méthodes ou algorithmes. Aménager une exception légale pour éviter ce genre de dérive n'est pas non plus une solution satisfaisante si l'on se réfère à la jurisprudence de l'OEB qui a en son temps vidé de sa substance l'exclusion de principe des logiciels. Il suffit pour cela de constater la multiplication de dépôts de procédés élémentaires, tels que le brevet one-click, du lien hypertexte ou d'inversion des couleurs !! Une telle main mise sur des techniques fondamentales de la programmation ne peut aller qu'à l'encontre de l'innovation, pourtant prônée par les défenseurs du brevet. C'est une crainte d'autant plus justifiée que la communauté du libre fonctionne essentiellement sur l'adaptation de principes récurrents aux programmes nouvellement ouverts. Un logiciel libre ne pourrait plus se développer en toute sécurité juridique si les éléments nécessaires à son architecture font l'objet d'un dépôt.

Enfin, il subsiste une incompatibilité⁵⁶ du brevet de logiciel avec les diverses licences libres. *«...Si, en conséquence d'une infraction au droit des brevets, [...] des conditions sont imposées au concessionnaire, qui entrent en contradiction avec les dispositions de la présente licence, elles n'exemptent pas le concessionnaire de respecter les dispositions de la présente licence »* En ce cas la licence perdure mais si aucune conciliation n'est trouvée entre les obligations légales et contractuelles alors il y a déchéance du droit de distribution. *« ...Si le concessionnaire ne peut procéder à la distribution de façon à satisfaire simultanément à ces obligations au titre de la présente licence et à toutes autres obligations pertinentes, alors, en conséquence de ce qui précède, le concessionnaire peut ne pas procéder du tout à la distribution du programme »* Donc un brevet qui irait à l'encontre d'une distribution libre du programme, obligerait le concessionnaire de la licence libre à s'abstenir complètement de distribuer le code ! La licence tombe d'elle-même.

2) Les risques d'une multiplication du contentieux.

⁵⁶ Notamment à l'article 7 de la GPL

Selon M. E.Di Filippo, « ...au sein de ce marché il existe une connivence malsaine entre les lawyers et les grandes firmes qui entretiennent une dynamique de contentieux à finalité concentrationniste »⁵⁷. Le brevet une fois obtenu ne peut être remis en cause qu'après une longue procédure. Dès lors, valable ou non, il constitue un moyen de pression auprès de sociétés concurrentes préférant transiger plutôt que d'assumer un procès long et coûteux⁵⁸. Insidieusement, cette multiplication des contentieux entraîne un accroissement des coûts de production dus tant aux frais judiciaires qu'au retard pris dans le développement du logiciel. Conséquence indirecte, mais tout aussi dommageable, les éditeurs risquent de se détourner de leur activité principale de réalisation de logiciels pour s'orienter vers celle de dépôt de brevets. La procédure de dépôt étant longue, cela retarde d'autant plus la date de mise sur le marché d'un logiciel, probablement devenu depuis lors obsolète.

Pour les auteurs de logiciels libres les risques de contrefaçon sont bels et bien réels. Ce qui aurait pour conséquence la désertion de nombreux programmeurs amateurs, formant pourtant le gros des troupes de la communauté, par peur de sanctions pénales.

Section 2 : La remise en cause du principe.

Très vite, alors même que le législateur venait à peine de soumettre le logiciel à la protection du droit d'auteur, s'est ouverte une autre voie prônant le droit des brevets comme alternative (I), notamment avec la décision Vicom du 15 juillet 1986⁵⁹. Restait donc à envisager l'étendue d'une telle protection (II).

I) L'alternative d'une protection per se.

Inexorablement, et malgré l'existence d'une forte opposition, notre droit tend à admettre la possibilité d'une brevetabilité des logiciels. Certes de manière frileuse et détournée dans un premier temps, il n'a pas fallu attendre très longtemps pour concevoir

⁵⁷ In mémoire DEA précité

⁵⁸ Cf l'affaire Adobe Vs Macromédia où l'exemple d'une transaction qui échoue et passe finalement en procès.

⁵⁹ CRT, Vicom, 15 juillet 1986, PIBD 1987, 409III, p134

un revirement de la logique juridique. C'est ainsi que le 20 février 2002, la commission européenne proposait un projet de directive allant dans ce sens.

A) De manière indirecte.

Un logiciel, en tant que tel, est exclu expressis verbis du champ du brevet (art 52.2 et .3 de la CBE). Aussi, un dépôt ne peut porter qu'indirectement sur un logiciel.

1) En tant que composant d'un procédé technique.

Ainsi, le procédé mis en œuvre par un logiciel peut faire l'objet d'un dépôt à la condition toutefois que la revendication porte sur le procédé et non le programme⁶⁰. Ce n'est pas la protection du logiciel en lui-même qui est recherchée mais plutôt son application à la conduite des étapes d'un procédé technique. C'est donc par le biais d'un artifice juridique qu'un logiciel, compris dans un ensemble plus vaste dont il n'est que l'un des composants, peut bénéficier des dispositions des articles L 611-1 et suivants du CPI. La reprise à l'identique d'un logiciel hors de son contexte technique et à des fins autres que celles revendiquées n'est pas susceptible de contrefaçon ce qui n'est pas sans poser de problème.

2) Par l'adaptation des critères de brevetabilité.

C'est pour tenter de répondre à cette problématique, que certains auteurs ont proposé de modifier les conditions d'accès au brevet afin d'englober les logiciels. En effet pour A. Casalonga⁶¹ « ...si l'objet revendiqué apporte une contribution de caractère technique à l'état de la technique, la brevetabilité ne devrait pas être mise en cause pour la simple raison qu'un programme d'ordinateur est impliqué dans sa mise en œuvre ». Cette adaptation n'est en somme qu'un retour aux principes fondamentaux. C'est-à-dire que ne serait pas exclu un genre particulier mais plutôt des objets répondant aux quatre critères suivants : que l'invention soit une information pure, traduise directement un phénomène naturel, manipule des idées abstraites ou apporte une solution à un problème purement mathématique. Il n'y a là aucune référence directe ou expresse à un logiciel.

⁶⁰ Décision Schlumberger de

⁶¹ A.Casalonga, in « la protection dans le domaine du logiciel en europe », Mélanges de Burst, 1996, p85

Cependant cette proposition se heurte, d'une part, à l'absence de réelle volonté politique en ce sens et d'autre part, à l'émergence d'un renversement de la donne juridique en faveur de l'adoption d'un régime pro-brevet.

B) Par un revirement de la logique juridique.

Dans un texte daté du 30 avril 1996 relatif aux lignes directrices pour les examinateurs en matière d'inventions liées aux programmes d'ordinateurs, le patent commissioner de l'USPTO s'estime prêt à considérer comme apte à la brevetabilité « *...tout computer implemented process* » dès lors que le procédé présente un impact dans le monde réel.

Par ailleurs un projet⁶² de directive européenne du 20 février 2002 marque une première étape vers la consécration législative des pratiques de l'OEB s'agissant d'invention mises en œuvre par ordinateur. Il n'était pas envisageable de refuser l'accès à la protection pour tout procédé industriel réalisé par un ordinateur car cela aboutirait à exclure du domaine de la brevetabilité de nombreuses inventions récentes. Pour paraphraser M. Macrez « *...le bit a remplacé le rouage* ». De lege ferenda un logiciel pourra être breveté. Mais est ce à dire qu'il jouira de la même étendue de protection que celle conférée par le droit d'auteur ?

II) Etendue de la protection du brevet de logiciel.

A) Appréhension de l'objet.

1) Par les revendications.

⁶²La Commission européenne a présenté une proposition de directive concernant la brevetabilité des inventions mises en œuvre par ordinateur. Il s'agit d'harmoniser les droits nationaux des brevets en ce qui concerne la brevetabilité des inventions mettant en œuvre un logiciel. De telles inventions peuvent d'ores et déjà être brevetées par l'Office européen des brevets (OEB) ou les offices nationaux des brevets, mais les modalités précises de la brevetabilité varient. Les entraves aux échanges de produits brevetés dans le marché intérieur resteront réelles tant que certaines inventions pourront être brevetées dans certains États membres et non dans d'autres. La proposition de directive sera présentée au Conseil de l'UE et au Parlement européen pour adoption au titre de la procédure de co-décision.

C'est par les revendications, et les descriptions jointes, que l'on détermine véritablement l'étendue de la protection. La question est donc de savoir comment décrire un logiciel. L'article L 612-5 CPI dispose que *« l'invention doit être exposée dans la demande de brevet de façon suffisamment claire et complète pour qu'un homme du métier puisse l'exécuter. »*

Il faut donc écarter une description au moyen du langage naturel, qui est peu précis et sujet à confusion. Or, comme le permet l'article R 612-13 CPI⁶³, l'intelligibilité et la concision doivent primer si la nature de l'invention l'exige. C'est pourquoi le recours au code source s'avère plus approprié⁶⁴ et non le code objet intelligible uniquement par la machine. Cette nécessité fait naître un paradoxe pour le moins surprenant. Le créateur cherchant une protection se verrait dans l'obligation de porter à la connaissance du public l'intégralité du code au même titre que l'auteur d'un logiciel libre ! Ce ne serait non plus sur le critère de la diffusion du code que se fonderait la différence avec les logiciels dits classiques mais sur la possibilité ou non d'en user librement.

Par ailleurs, et pour accroître d'autant plus le trouble déjà latent, il est fort probable que la revendication d'un brevet de logiciel se fasse plutôt en pseudo code, c'est-à-dire sous forme enrichie d'annotations. Elle est donc plus complète et utile que la simple diffusion du code source prévue par le logiciel libre !!

2) Par sa délimitation avec les méthodes et les algorithmes.

Pour avoir une nature technique, un logiciel doit produire un effet technique. Or, c'est la conception de l'algorithme engendrant la création de cet effet qui permet la mise en œuvre de la solution technique. Il est donc à craindre que ce soit la seule protection de l'algorithme qui est recherché par le dépôt. Pour M. Macrez⁶⁵ *« ...breveter un logiciel amène à retenir que la protection porterait sur la fonction technique, le principe abstrait dans son rapport avec une application déterminée, c'est-à-dire l'implémentation de l'algorithme »*. Nous touchons là le vrai problème de la nature de l'algorithme. Ce que les juristes analysent comme étant une seule notion, les informaticiens l'appréhendent selon sa double nature : dans un premier temps comme une approche intellectuelle abstraite, puis ensuite comme sa traduction en un langage

⁶³ **R 612-13** : La description est présentée dans les conditions [...] à moins que la nature de l'invention ne permette une présentation différente plus intelligible et plus concise.

⁶⁴ cf R 612-13 1°

⁶⁵ M. F. Macrez, in document de travail sur le projet RNTL

donné. Il s'agit d'une certaine formalisation, à l'instar de l'idée prenant corps en droit d'auteur. Le même algorithme peut donc faire l'objet de brevets différents en fonction de son application. La frontière devient alors particulièrement perméable, et il est illusoire de croire en l'efficacité d'une exclusion légale tant il est difficile d'apprécier le rattachement de l'algorithme à une application concrète déterminée qui elle est brevetable. A l'image des premiers brevets sur le logiciel, l'algorithme serait pas protégé, *in natura rerum*, mais seulement en tant qu'il assure une fonction technique lors de son intégration à un procédé breveté dans son ensemble. Autrement dit, on tend vers toujours plus de protection par le jeu de l'accession. Cette extension de la propriété industrielle se fait au détriment de la propriété littéraire et artistique.

B) Vers un cumul des protections ?

1) Un choix entre coexistence et cumul.

Opter pour la brevetabilité implique que soient définies les règles de partage avec la protection accordée au titre du droit auteur. La coexistence suppose deux objets de droit distincts ce qui ne reflète pas la problématique propre au logiciel. Il serait plus adéquat d'envisager un cumul qui se traduirait par la protection par le brevet des aspects techniques sous jacents tandis que leur concrétisation littéraire sous forme de code source toucherait au droit d'auteur. Néanmoins, légalement, cette approche se heurte au principe de non cumul instauré par la loi, dès lors que l'invention est simplement brevetable. En outre, l'application de la théorie de l'unité de l'art, qui sous entend que forme et fonction sont dissociées, reste en la matière hasardeuse.

Il apparaît alors nécessaire d'aménager les dispositions relatives à l'interdiction du cumul afin d'organiser au mieux une telle cohabitation.

2) En matière de contrefaçon.

Les prérogatives du titulaire d'un brevet sont strictement encadrées par l'article L 613-3 CPI à partir duquel sont interdites à défaut du consentement du créateur, la fabrication et la mise sur le marché. En revanche, confrontée au droit d'auteur, la marge de l'utilisateur d'un logiciel est restreinte par le peu d'exception en sa faveur (copie de sauvegarde, décompilation...). De plus, seule l'utilisation de moyens équivalents à ceux

revendiqués est préjudiciable. Le brevet de résultat étant prohibé, l'obtention d'un résultat similaire par des moyens autres n'est pas une contrefaçon.

Le brevet de logiciel offre alors une protection moindre eu égard au plus faible nombre d'actes considérés comme répréhensibles.

TITRE II : UNE PROTECION AD HOC.

CHAPITRE I : Le recours au contrat.

Le licencié reçoit le droit de copier, de distribuer et de modifier le logiciel. Or, il n'y a pas eu transfert de droit, dans le sens où l'entend le CPI. Ce n'est plus sur le terrain du droit de la propriété intellectuelle qu'il faudrait raisonner mais sur celui du droit des obligations. Dans cette optique le contrat de licence, qui peut revêtir plusieurs formes (section 1), prime les droits d'auteurs.

Section 1 : Typologie des diverses licences.

Le monde des logiciels libres n'offre pas un visage homogène⁶⁶. Par le truchement des licences, parfois incompatibles entre elles, il est possible d'en adapter conventionnellement le régime. Le logiciel libre est né de la volonté de ne pas laisser un seul type de logiciel s'imposer aux utilisateurs. Les licences sont un mécanisme par lequel le titulaire des droits sur le logiciel libre reconnaît à un tiers le droit de le modifier, le copier ou le diffuser. Néanmoins, ce n'est nullement une abdication par le titulaire de ses droits d'auteur. Le système des licences accorde contractuellement certains droits, mais en fixe parallèlement les limites.

Il existe essentiellement trois grandes catégories de licences qui régissent les logiciels libres. Celle de licence libre stricte, dont la principale représentante est la Licence Publique Générale GNU, celle de type domaine public, et enfin celle semi-libre, à laquelle se rattache l'opensource.

Copier et adapter librement les logiciels constitue la quintessence de la philosophie du libre. Il s'agit du dénominateur commun à toutes les licences freesoftware. Mais à la périphérie de ce noyau central, chaque type de licence apporte son particularisme.

⁶⁶ Exemple de licences : BSD license, MIT license, QtP, CVW, GPL, Apache license, Mozilla, Sleepycat, APSL, IBM public license, Zlib, SISSL, MPL 1.1, Vovida, Artistic, Ricoh SCPL, IOSL...etc

I) Les licences ouvertes.

A) Les licences de domaine public.

Elles offrent les mêmes prérogatives que les autres licences mais se différencient par l'absence d'obligation de pérennisation de la licence à travers les évolutions du logiciel. La publication du code source après chaque nouvelle version n'est pas exigée. Dès lors rien n'empêche le développement d'un logiciel propriétaire à partir d'une mouture libre. On pourrait alors craindre qu'une modification minimale du logiciel originel ait pour conséquence son appropriation. Le dépôt d'un copyright classique sur l'ensemble permettrait la mise sur le marché de variante commerciale de logiciel à l'origine libre. Certains auteurs, tels M. Y.Bailly ou Mme Clément-Fontaine⁶⁷, ont avancé l'idée que l'auteur pouvait agir sur le fondement de son droit moral au respect de son œuvre. Or, en matière de logiciel, celui-ci est inexistant. Il est toujours possible d'avancer que ce droit perdure lorsqu'il y a atteinte à l'honneur ou à la réputation de l'auteur. Mais s'agit-il en l'occurrence d'un tel préjudice ? Une telle conception impliquerait une étude au cas par cas, selon un point de vue subjectif. Le préjudice serait en effet évalué en fonction de la notoriété de l'auteur, et quel serait le critère employé pour définir si la modification est substantielle ou non ? Il serait peut-être plus judicieux de faire appel à des mécanismes de droit civil, comme la responsabilité délictuelle.

B) Les licences semi-libres

L'autre grande catégorie est celle des semi-libres, avec comme axe majeur l'opensource. La société Netscape en a été l'instigateur, grâce à sa Mozilla Public Licence (MPL). C'est une variante de cette licence que la société exploitant le DivX a opté pour diffuser son code source⁶⁸. Elles correspondent à un juste milieu entre d'une part, la licence libre stricte qui oblige lors de l'intégration d'un logiciel libre à un produit commercial la transmission du code source issu de cette fusion, et d'autre part, celle du domaine public, qui comme nous l'avons vu, encours le risque d'une

⁶⁷ Précités

⁶⁸ Cf annexe n°5

appropriation. Par le biais de l'open source, il est concevable de créer à partir d'un logiciel libre, dont le code source sera toujours disponible, une nouvelle œuvre logicielle avec des modifications propriétaires. L'exploitation commerciale s'en trouve alors facilitée.

II) Les licences de type GNU.

La licence la plus communément rencontrée sur Internet est la Licence Publique Générale (GNU GPL). Acronyme récursif elle signifie Gnu is Not Unix. Elle gère la majorité des logiciels libres, comme le plus célèbre d'entre eux Linux⁶⁹. Le but de cette licence est, pour la citer, *"de donner à tous les utilisateurs la liberté de redistribuer et de modifier les logiciels GNU"*⁷⁰. Pour cela, un copyright un peu particulier est apposé sur chaque version du logiciel et permet des droits illimités d'usage. Ce copy...right, devenu par "gauchisation" copyleft, est transmis systématiquement à tous les utilisateurs futurs et répercuté aux versions ultérieures utilisant tout ou partie d'un logiciel GNU.

M. Bailly précise ⁷¹ que « pour éviter leur appropriation par des tiers, les logiciels libres régis par cette licence ne sont pas dans le domaine public, mais sont protégés par une forme particulière de copyright, le copyleft. ». Cela reste contestable dans la mesure où une fois dans le domaine public, une œuvre est insusceptible d'appropriation. Les licences ont plutôt pour objet d'organiser la liberté octroyée aux utilisateurs, sans qu'il soit possible à quiconque de s'approprier effectivement le logiciel. Non seulement la licence s'avère appropriée à la protection du logiciel, mais est aussi un vecteur remarquable de développement.

⁶⁹ Dont la mascotte est le célèbre Tux le pingouin inventé par Linus Torvald

⁷⁰ Licence accessible sur www.linux.org

⁷¹ article publié sur www.net-iris.com

Section 2 : Leur appréhension par le droit commun des contrats.

I) L'encadrement d'une communauté.

A) Un lien contractuel unioniste.

Les logiciels libres se fondent sur un mode de développement coopératif et volontaire sans rapport de force permettant de fédérer les égoïsmes individuels hors de toute structure coercitive. L'ensemble de cette communauté est régi à travers la licence qui organise cet encadrement unilatéralement. Elle s'entend en dehors de toute discrimination de personnes ou de domaine d'application. La communauté se rassemble autour d'une licence qui édicte les normes à suivre. Celle-ci n'est pas négociable et est donc un contrat d'adhésion. Le licencié adhère sans condition aux conditions établies au présent contrat. Mais le cessionnaire également car il ne fait que reprendre les termes de la licence sous laquelle il désire diffuser son logiciel ce qui l'apparente à un contrat type.

Ne transférant aucun droit réel, la licence ne peut être qualifiée de contrat de vente. Si en revanche elle concède un droit personnel alors il nous faudra parler de licence au sens que l'entend le CPI pour les brevets. La gratuité n'est pas un critère déterminant à la qualification de ce lien car un logiciel libre peut être concédé à titre gratuit comme onéreux. Un logiciel fondé sur la gratuité de sa diffusion sera nommé Freeware, sans pour autant que son code source soit livré. Reste alors à lui trouver une catégorie de contrat civil nommé auquel une qualification en contrat Sui Generis s'imposera.

L'opensource est né d'une vision plus mercantile de la licence GNU⁷². Mais elle présente un double visage. La licence reste gratuite tant que sont respectées les finalités contractuellement prévues. En revanche, la question reste posée lorsque le code source est employé à des fins commerciales. Dans la plupart des cas, la licence n'y fait pas expressément référence, mais elle impose l'obtention du consentement des auteurs pour

⁷² cf l'article de M.Perens sur www.linux.org

toute autre utilisation du code. Il est donc fort probable qu'en contrepartie de cet accord une rétribution financière soit réclamée, par l'entremise d'un contrat ad hoc. L'opensource permet cette dualité, et fait même de la coexistence de logiciels libres et commerciaux sur des variantes du même code source son principal intérêt. Le régime juridique est donc fonction de l'utilisation faite du logiciel, d'une part, celui d'un contrat de prêt à usage pour la mise à disposition gratuite du code, et d'autre part, de contrat de licence quand la diffusion du code est faite à titre onéreux. La protection juridique dépend donc de la destination du logiciel et non de sa qualification intrinsèque. Le choix d'un contrat de prêt à usage ne fait toutefois pas l'unanimité en ce qu'il oblige à la restitution de la chose. Pour Mme Clement-Fontaine, cela importe peu du fait du caractère immatériel du logiciel libre. Pour une distribution commerciale et en l'absence de transfert de droit, la qualification en contrat de louage de chose incorporelle⁷³ ne soulève aucun problème particulier.

B) La formation du contrat.

De par leur nature contractuelle, les diverses licences n'échappent pas aux exigences légales existantes en la matière⁷⁴. Les dispositions de l'article 1108 du code civil ainsi que toutes celles relatives au droit des obligations trouvent à s'appliquer. Les licences libres n'ont de véritablement spécifique que leur objet, le logiciel⁷⁵.

Une fois les conditions préalables de formation remplies, il convient de s'attarder aux spécificités des contrats de licences. L'une d'elle relance le débat né de la pratique de "shrink wrap licence". Le consentement du licencié est-il suffisamment éclairé si sa volonté n'est pas clairement exprimée ? En la matière, le principe du consensualisme prévaut et le consentement peut être donné sous une quelconque forme tant qu'il répond de manière claire et précise à l'offre de contracter. Cependant, le contrat est souvent conclu en ligne. La volonté peut-elle alors se manifester au travers d'une simple checkbox⁷⁶ ? Beaucoup⁷⁷ en doute relevant le manque de conscience des

⁷³ **Art 1709** code civil : *Le louage des choses est un contrat par lequel l'une des parties s'oblige à faire jouir l'autre d'une chose pendant un certain temps, et moyennant un certain prix que celle-ci s'oblige de lui payer.*

⁷⁴ Pour une étude plus approfondie, se reporter au travail de Mélanie Clement Fontaine sur la licence GNU, mémoire DEA Montpellier 1999.

⁷⁵ Pour ce qui le concerne Cf le Titre I du présent document

⁷⁶ Checkbox = case à cocher

⁷⁷ O.Gnos, M.Clement Fontaine ou encore M.Vivant, in Lamy droit de l'informatique et des réseaux.

internautes lorsqu'ils valident l'opération instinctivement, sans en conclure pour autant à une invalidité de principe. La réponse relève donc de la casuistique.

II) L'exercice de la liberté à travers le contrat.

A) Droits et obligations du licencié.

« *Les activités autres que la duplication, la distribution et la modification ne sont pas couvertes par la présente licence* »⁷⁸ Quid du droit d'utilisation ? En toute logique

1) Le fonds commun.

Selon E. Moglen, L'objet d'une licence est « *de rendre le logiciel libre en créant un fonds commun auquel chacun peut ajouter, mais duquel personne ne peut retrancher* »⁷⁹. Pour cela le licencié doit bénéficier de quatre grandes libertés fondamentales : celle d'exécuter le programme pour tous les usages, celle d'étudier le fonctionnement du programme et de l'adapter, celle de redistribuer des copies et celle d'améliorer le logiciel. A partir de là les différentes licences divergent. Certaines autorisent une appropriation du code dans un logiciel propriétaire, par exemple la licence Apache, d'autres l'interdisent comme la GPL. Pour R. Stallman « *...[la licence] garantit que vous avez la liberté d'étudier, de modifier et de redistribuer un programme, en affirmant que personne n'est autorisé à vous retirer ces libertés en le redistribuant* »⁸⁰.

2) Les effets du contrat.

De par sa nature de contrat synallagmatique, le contrat de licence fait naître à la charge des parties des obligations réciproques. Celui qui accorde la licence doit mettre le licencié en situation d'user de la plénitude de ses droits. Cette liberté, loin d'être absolue, est encadrée par l'obligation de respecter la libre utilisation du logiciel, le

⁷⁸ Art 1 alinéa 2 de la licence GNU GPL

⁷⁹ Eben Moglen, <http://moglen.law.columbia.edu>

⁸⁰ R.Stallman, la licence GNU GPL et l'américain way, in multitudes, mai 2001

préambule de la licence GNU dispose⁸¹ que «...*Pour protéger vos droits, nous devons apporter des restrictions, qui vont interdire à quiconque de vous dénier ces droits, ou de vous demander de vous en désister...* ». Ainsi toute distribution du logiciel s'effectue selon certaines modalités, à savoir qu'il doit être transmis avec la licence, avec un minimum d'indications relatives aux droits d'auteur et avec l'intégralité du code source exécutable.

Il n'existe pas de relation contractuelle entre le licencié et le destinataire. Ce dernier reçoit les mêmes prérogatives que le licencié⁸² par le jeu de la licence mais ne peut voir sa liberté restreinte par une quelconque obligation contraignante imposée par lui. N'étant pas titulaire des droits, il ne peut, en vertu du principe *nemo plus juris, retoucher* la licence à son profit.

Il est à noter que par l'effet relatif des contrats⁸³, une licence n'a d'effet qu'inter partes et non à l'égard des tiers. A ce titre les sous licences sont interdites sans l'accord du titulaire des droits sur le logiciel, comme le précise l'article 4⁸⁴ de la GPL : «*Le concessionnaire ne peut dupliquer, modifier, concéder en sous-licence ou distribuer le programme, sauf si cela est expressément prévu par les dispositions de la présente licence. Toute tentative pour autrement dupliquer, modifier, concéder en sous-licence ou distribuer le programme est réputée nulle, et met automatiquement fin aux droits du concessionnaire au titre de la présente licence* »

Cependant, rien n'exclu l'existence de prestations jointes, en relation avec l'objet de la licence. C'est le cas notamment des contrats passés avec les éditeurs de distributions Linux⁸⁵. Il s'agit de contrats distincts, connexes mais ne dépendant pas directement de la licence. Celle-ci ne traite pas des modes d'exploitation ou des services à valeur ajoutée. Ce contrat qui peut être lui-même une nouvelle licence participe du renouvellement perpétuel du cycle des logiciels libres.

⁸¹ *To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights*

⁸² *if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights*

⁸³ Art 1165 cc

⁸⁴ *you may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance*

⁸⁵ Mandrake, Red hat, Suze, Debian... etc

B) Les limites.

1) Un manque de conformité au droit.

L'essentiel des licences disponibles est rédigée en anglais. Aussi, la plupart du temps, et malgré l'existence de traduction, seule la version en langue anglaise fait foi (comme c'est le cas pour la GPL). En droit Français, la loi Toubon impose qu'un contrat d'adhésion soit rédigé dans la langue de Molière. Se pose alors la question de la validité de la licence. Un juge français saisi sur ce point serait en droit d'annuler la licence. C'est pourquoi la FSF, en collaboration avec Mme. Clément-Fontaine, tente d'élaborer une version française de la GPL.

L'exploitation d'une licence libre soulève un sérieux problème de protection du consommateur. La licence GNU, et dans une moindre mesure celle opensource, inclut régulièrement des clauses d'exonération de responsabilité. En effet, parce que l'utilisation du logiciel est libre et gratuite, aucune garantie n'est fournie. Les détenteurs des droits sur le logiciel fournissent le programme en l'état, sans aucune sorte de garantie explicite ou implicite. Le consommateur assume dès lors tous les risques quant à la qualité du produit. La totalité des coûts pour la remise en l'état conforme d'un logiciel défectueux lui incombe. Cette absence de garantie a pour conséquence d'exonérer de sa responsabilité l'auteur du fait des dommages liés aux dysfonctionnements de son logiciel. Ces dispositions sont donc contraires à la directive européenne du 25 juillet 1985 sur la protection des consommateurs. Toutefois ces licences s'adressent à un public averti et non dénué d'expérience. Un simple avertissement du caractère hautement technique de l'utilisation de ces licences et un encadrement strict des indemnisations envisageables permettraient de remédier au manque de garantie et ne pas tomber ainsi sous le coup de la réglementation sur les clauses abusives. Toutefois, dans la mesure où de nombreux éditeurs de distributions de logiciels libres jouent de facto un rôle de certificateur de solutions leur responsabilité devrait pouvoir être engagée.

Dans certaines licences, il est stipulé que toute incorporation du code dans un travail plus large (le contrat parle de «larger work»⁸⁶) est autorisé à la condition que ce travail ait pour vocation une utilisation personnelle ou une ingénierie inverse⁸⁷. Cette restriction a pour but de contrôler la parution de programme intégrant le code. Les termes de cette clause se rapprochent, dans l'esprit, d'une clause de non de concurrence, en vertu de laquelle l'utilisation de l'œuvre est libre si elle n'a pas notamment pour conséquence de créer un préjudice injustifié aux intérêts commerciaux légitimes de l'auteur.

2) Le lien viral.

Un logiciel libre ne peut être incorporé que dans un logiciel qui garantit les mêmes libertés de redistribution gratuite. Ce principe viral infecte toute œuvre dérivée. L'auteur de l'œuvre composite ne peut ainsi réclamer une rémunération au titre de la distribution du code source. Pour y remédier, il doit s'assurer que ses propres éléments sont indépendants et fassent l'objet d'une licence de logiciel séparée. L'utilisation marginale du code fait tomber l'ensemble dans le libre. Il est à noter que ce n'est pas le cas de toutes les licences libres, la GPL étant la plus restrictive à ce sujet. La licence OpenSource en son art 9 impose l'absence de contamination d'autres logiciels distribués avec le programme. Mais cela ne résout que partiellement le problème des œuvres dérivées car, a contrario, l'article 3 exige que leur distribution soit soumise aux mêmes conditions que celles qui protègent le programme original. Lorsque l'on parle de dérivation directe en aval alors effectivement toutes les licences sont virales. Mais en amont, c'est-à-dire quand un logiciel libre est cette fois-ci intégré, la contagion n'est pas certaine. En ce sens la GPL connaît une version allégée, la LGPL, qui permet l'intégration de modules non libres.

Qu'advient-il alors de la réunion de composants originaux en un unique programme exécutable compilé ? C'est sur ce problème qu'aura à statuer le tribunal de Boston (Massachusetts, USA) dans l'affaire MySQL AB contre Progress Software. Une injonction préliminaire a déjà été rendue le 1^{er} mars 2002 pour ce qui concerne le conflit sur la marque mais rien encore à propos des droits de distribution sous licence GNU GPL. Y a-t-il infraction lorsque les codes ont été distribués commercialement sans l'accord du cessionnaire ? Pour MySQL AB un préjudice a été causé à la communauté

⁸⁶ Par exemple la licence Open DivX (www.divx.com)

⁸⁷ article 4a «you may incorporate the code into a larger work and distribute that larger work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications »

libre. En l'absence de jurisprudence en l'espèce, cette décision permettra sûrement d'apprécier la valeur juridique de la licence.

CHAPITRE II : La cathédrale et le bazar : l'appropriation de l'information.

« ...Je pensais que les logiciels les plus importants (comme les systèmes d'exploitation et les très gros outils comme Emacs) devaient être conçus comme des cathédrales, soigneusement élaborés par des sorciers isolés ou des petits groupes de mages travaillant à l'écart du monde, sans qu'aucune version bêta ne voie le jour avant que son heure ne soit venue.

Le style de développement de Linus Torvalds - distribuez vite et souvent, délégez tout ce que vous pouvez déléguer, soyez ouvert jusqu'à la promiscuité - est venu comme une surprise. À l'opposé de la construction de cathédrales, silencieuse et pleine de vénération, la communauté Linux paraissait plutôt ressembler à un bazar, grouillant de rituels et d'approches différentes (très justement symbolisé par les sites d'archives de Linux, qui acceptaient des contributions de n'importe qui) à partir duquel un système stable et cohérent ne pourrait apparemment émerger que par une succession de miracles.

Le fait que ce style du bazar semblait fonctionner, et bien fonctionner, fut un choc supplémentaire. Alors que j'apprenais à m'y retrouver, je travaillais dur, non seulement sur des projets particuliers, mais encore à essayer de comprendre pourquoi le monde Linux, au lieu de se disloquer dans la confusion la plus totale, paraissait au contraire avancer à pas de géant, à une vitesse inimaginable pour les bâtisseurs de cathédrales... »

Eric S. Raymond in « la cathédrale et le bazar »

Nous avons déjà développé la thèse, communément admise, selon laquelle l'information est de libre parcours. Néanmoins et nonobstant ce qui a été affirmé, les détracteurs du logiciel libre cherchent à endiguer cette diffusion anarchique du savoir en prônant sa réservation (section 1). Or, l'existence et le développement des logiciels libres sont inhérents à la circulation du savoir (section 2).

Section 1 : La cathédrale, l'idée du tout appropriable.

S'appropriier le savoir apparaît comme une nécessité lorsque entre en jeu d'énormes intérêts financiers. Posés en gardes fous contre les dérives privatistes, les dispositions du Code de la propriété intellectuelle s'opposent à une telle réservation. C'est donc par des moyens détournés que tente de s'imposer un "capitalisme cognitif" (I) inapplicable aux logiciels libres (II).

I) Les différents modes de réservation.

A) Le prix du savoir

Microsoft dans son rapport «Shared source» du 3 mai 2001, présentait la licence GPL comme «*découragement au développement de logiciel commerciaux, générateur de valeur économique, en termes de propriété intellectuelle [...] Cette licence réduit, voire annihile la relation symbiotique qui existe entre la recherche académique et gouvernementale et l'ensemble de la communauté commerciale.*».

La possibilité de reproduire indéfiniment et à un coût dérisoire la connaissance rend inopérantes les mesures visant à contraindre les consommateurs à payer. Cette connaissance n'a intrinsèquement plus de valeur commerciale.

B) La protection du savoir

Le droit n'admet pas une appropriation de principe de l'information par l'existence d'un droit privatif général. C'est à travers des régimes spécifiques et dérogatoires que peut apparaître une appréhension juridique du savoir. Plus que son contenu même, c'est la protection de sa valeur qui est recherché. Les atteintes à la confidentialité de l'information sont ainsi pénalement réprimées par les délits d'initiés, du secret professionnel ou encore par la loi Godfrain relative à l'accès frauduleux dans un système automatisé de données. Toutefois, elle ne peut faire l'objet d'une quelconque soustraction. L'information n'a, dans cette optique, d'intérêt que si elle

n'est pas communiquée. Elle devient un bien immatériel consommable au premier usage.

Du point de vue du droit civil, les pratiques anticoncurrentielles sont sanctionnées. Le détenteur d'informations sensibles peut agir en responsabilité délictuelle en cas d'agissement parasitaire ou de concurrence déloyale.

Les éditeurs de logiciels propriétaires font valoir l'importance du secret autour d'algorithmes, codes et méthodes, véritables richesses de leur société. La diffusion de ces éléments constitue donc un préjudice alors même qu'ils ne sont pas protégeables en l'état par la propriété intellectuelle. Une note de service de Microsoft connue sous le nom de « halloween »⁸⁸ fait état des pertes occasionnées pour l'entreprise par la diffusion d'informations sous le modèle libre.

II) L'applicabilité aux logiciels libres.

Pour expliquer en quoi l'appropriation du savoir allait à l'encontre de toute logique de communauté E. Moglen⁸⁹ créa deux personnages. Le premier, l'IPdroïde qui « *...sait tout sur quelque chose et rien sur tout le reste...* », est une métaphore de l'incapacité de la propriété intellectuelle à faire sa révolution. Elle engendre la complexité lorsqu'elle est confrontée au monde analogique. La propriété protège alors les investissements et la distribution des supports physiques. Le numérique renverse la logique et rend la complexité inutile voire paralysante. L'instabilité chronique de ses différents régimes entraîne l'incertitude et amène à adopter un régime ad hoc aux nouveaux objets de droits⁹⁰. Chacun cherche à faire valoir le régime qui lui est le plus favorable sans tenir compte d'une quelconque cohérence juridique ce qui conduit à l'effondrement des paradigmes.

Le second personnage est l'économain. Il se fonde quant à lui sur la motivation des individus. Pour lui le logiciel libre détruirait toute motivation mercantile et vénale. Or, la principale récompense pour l'auteur d'un logiciel libre réside dans la renommée et la reconnaissance qu'il peut tirer de son travail. Et seulement, à partir de là, un intéressement financier peut parfois en découler. Pour certains même comme E. Raymond il s'agit « *du boost de l'ego* »⁹¹.

⁸⁸ V. Valloppillil, <http://www.opensource.org/halloween.html>

⁸⁹ E. Moglen, in *l'anarchisme triomphant*, 1999, <http://emoglen.law.columbia.edu>

⁹⁰ Comme par exemple le régime particulier des Bases de données ou du logiciel

⁹¹ auquel il ajoute une autre comparaison pseudo ethnographique de la composition du logiciel libre et du potlach chez les indiens Kwkiutl. E.S. Raymond, in *homesteading the noosphere*, <http://www.tuxedo.org>

Même si nous émettons des doutes sur la pertinence de l'enseignement de l'IPdroïde, force est de reconnaître que celui de l'économain n'est pas dénué de tout fondement. Lorsque les réseaux informatiques permettront la rémunération directe des auteurs en faisant transiter de petites sommes d'argent, le modèle libre trouvera alors un moyen efficace pour récompenser ses contributeurs. Comme le souligne R. Stallman⁹² « ...si vous aimez un livre et qu'un menu contextuel vous invite à cliquer ici pour envoyer un dollar à l'auteur, ne cliqueriez vous pas ? »

Section 2 : Le bazar, la circulation du savoir.

I) Principe fondateur de la communauté du libre

L'intérêt d'une communauté réside dans la capacité à réunir autour de projets complexes une multitude de participants dont les efforts, du plus modeste au plus fondateur, servent la même cause. Cette fusion nécessite que chacun puisse bénéficier de l'apport de l'autre. Or, la facilité avec laquelle l'information peut être reproduite augmente d'autant plus sa diffusion et du coup sa vocation à être communiquée.

Le logiciel libre repose sur ce postulat d'échange de l'information sans entrave. La libéralisation du code source en constitue le fondement. En ce sens, il se distingue d'autres types de logiciels offrant pourtant une liberté similaire. En premier lieu, le logiciel libre n'est pas un freeware, ce qu'une traduction littérale aurait pu laissé entendre. Un freeware est un logiciel gratuit, mais l'absence de contrepartie financière ne suppose pas que l'utilisateur puisse modifier et distribuer l'œuvre à son gré. Ce n'est pas non plus en shareware. Certes, il est possible de diffuser et de reproduire à titre gratuit l'œuvre, cependant cette autorisation n'est que temporaire et toute modification constitue un acte de contrefaçon. Il en va de même pour le crippleware qui n'est qu'une version bridée de l'original. Dans toutes ces hypothèses, le code source n'est pas disponible. Quid alors de l'abandonware ? Il s'agit là de logiciels anciens (souvent des jeux), depuis longtemps sortis du circuit commercial. Certains sites⁹³ les font revivre

⁹² R.Stallman, in la liberté ou le copyright, Multitudes mai 2001.

⁹³ Consulter les sites tels que www.abandonware-france.org , www.web-emulation.com , www.bhlegend.com ou encore www.oldiz.com

parfois dans l'illégalité⁹⁴. Lorsque l'éditeur donne son aval⁹⁵, le logiciel devient disponible au téléchargement gratuitement, ce qui le rangerait plutôt alors du côté des freeware. Toutefois, dans certains cas le code source est lui aussi en partie ou totalité distribué⁹⁶ afin que les internautes puissent proposer des patches, mods, nouveaux niveaux... Est-ce à dire qu'il s'agit là de logiciels libres ? La réponse n'est pas évidente dans la mesure où l'accès au code source constitue la pierre angulaire du libre. En son absence⁹⁷ l'abandonware n'est pas un logiciel libre. Dans le cas contraire, il apparaît nécessaire de distinguer si la diffusion se fait par le biais d'une licence libre quelconque, ou non. Dans le premier cas il s'agira d'un logiciel libre, dans l'autre cas d'une œuvre tombée dans le domaine public.

II) L'applicabilité du modèle libre à d'autres créations.

A) Une réalité hors de l'univers numérique.

Les licences, comme celle d'art libre⁹⁸, sont nées de l'observation du monde du logiciel libre et de l'Internet, mais leur domaine d'application ne se limite pas au numérique.

Il est dès lors possible de mettre une peinture, un roman, une sculpture, un dessin, une musique⁹⁹, un poème, une installation, un film¹⁰⁰, une recette de cuisine, un site web, enfin bref, toutes créations qui peuvent se réclamer d'un certain art sous une forme libre

Rendre une œuvre littéraire libre¹⁰¹, par exemple, signifie que son contenu l'est. En d'autres termes il devient gratuit et distribuable et qu'importent les modifications qui y ont été apportées, sous la réserve unique d'y adjoindre la version originale. C'est dans la volonté de partage des idées que cette démarche s'inscrit.

Placer le contenu d'une œuvre dans le modèle libre implique que sa copie, sa distribution, sa modification et sa traduction soient possibles sans autorisation expresse de l'auteur. La copie de l'œuvre est soumise à la condition de respecter le droit à la

⁹⁴ Lorsque les éditeurs n'ont pas donné leur accord. Le fait que les logiciels ne soient plus dans le commerce ne supprime pas les droits d'auteurs à leur égard !

⁹⁵ Comme la défunte société Lankhor, www.lankhor.net

⁹⁶ Comme par exemple ceux de doom, quake, flight simulator...etc

⁹⁷ C'est-à-dire pour la plupart des abandonware.

⁹⁸ Disponible sur www.artlibre.org

⁹⁹ <http://www.musique-libre.com>

¹⁰⁰ <http://www.videontv.org/licence>

¹⁰¹ http://www.kitetoa.com/les_livres/roman/sommaire_roman.html

paternité de l'auteur ainsi que l'obligation de reproduire la licence libre qui s'y attache. Le droit à modification est en revanche plus encadré. Il nécessite que soit fait mention dans l'œuvre dérivée de toutes transformations et ajouts apportés à l'œuvre primaire, transmise également. En outre les modifications tombent sous le coup de la licence et l'œuvre composite doit en respecter les dispositions. Enfin, la traduction, à défaut d'autorisation de l'auteur, reste possible si elle est accompagnée de l'œuvre originale et avec l'indication de l'absence de l'accord.

De ces principes fondamentaux, il ne faut pas en conclure que le licencié a tous les droits. En dehors des cas prévus par la licence, l'auteur premier reste maître de son œuvre. Ce n'est pas la négation pure et dure du droit d'auteur bien au contraire. C'est même dans son cadre que ces licences s'émancipent. Leur préambule est à ce titre révélateur « ... *Loin d'ignorer les droits de l'auteur, cette licence les reconnaît et les protège. Elle en reformule le principe en permettant au public de faire un usage créatif des œuvres d'art. Alors que l'usage fait du droit de la propriété littéraire et artistique conduit à restreindre l'accès du public à l'oeuvre, la Licence Art Libre a pour but de le favoriser.*

L'intention est d'ouvrir l'accès et d'autoriser l'utilisation des ressources d'une oeuvre par le plus grand nombre. En avoir jouissance pour en multiplier les réjouissances, créer de nouvelles conditions de création pour amplifier les possibilités de création. Dans le respect des auteurs avec la reconnaissance et la défense de leur droit moral... »¹⁰²

B) Mais une illusion pratique.

Néanmoins cette pratique ne peut être que marginale. En effet, la généraliser implique une refonte totale des processus classique de diffusion des œuvres. Les enjeux économiques sont par trop importants pour qu'en l'absence du concours des grandes majors du secteur il soit difficile d'étendre la gauche d'auteur au-delà de cas atypiques. Certes beaucoup plus adapté au monde numérique, les licences libres semblent marquer le pas dans l'univers analogique et matériel. L'avantage incontestable du logiciel libre est de bénéficier d'un support de partage du savoir qui rend aisé la modification et la

¹⁰² Préambule de la licence d'art libre disponible sur le site www.artlibre.org

transmission des œuvres dérivées libres. Avec les supports physiques, l'échange est plus compliqué.

Malgré cela, beaucoup d'œuvres sont transposables sur Internet par leur numérisation (ouvrages, musiques, vidéo...). Le logiciel est une œuvre cachée en ce sens qu'elle n'est accessible que par l'intermédiaire de son code source. Or, il n'est pas nécessaire d'en avoir connaissance pour pouvoir utiliser le logiciel. En revanche, il est indispensable à l'exercice des droits accordés par la licence. C'est pourquoi, cette dernière organise sa libéralisation. Ce problème ne se pose pas pour les autres types d'œuvres. Elles n'ont fait que changer de support. Une fois noyées dans un océan de 0 et de 1, leur modification en devient même plus facile ! Leurs licences ne font somme toutes qu'aménager les droits patrimoniaux qui y sont attachés.

TABLE DES MATIERES.

INTRODUCTION.....	5
TITRE I : LE LOGICIEL LIBRE ENCADRE PAR LE CODE DE LA PROPRIETE INTELLECTUELLE.....	9
CHAPITRE I : EN TANT QUE CREATION DE L'ESPRIT.....	9
<i>Section 1 : Une réservation par assimilation</i>	<i>10</i>
I) Le logiciel, une œuvre littéraire.....	10
A) Le choix d'une réservation contre nature.	10
B) L'objet de la protection.	11
1) Le caractère atypique du logiciel libre.....	11
2) Un logiciel polymorphe.	12
II) Recours au régime de droit commun.	12
A) Une genèse perpétuelle ?	13
1) Une œuvre inachevée ou dérivée ?	13
2) Une forme originale.	15
B) Des prérogatives diminuées.	15
1) Un abandon des droits patrimoniaux.	16
2) La persistance des droits moraux.	17
<i>Section 2 : les dérives de la propriété littéraire et artistique.....</i>	<i>17</i>
I) La difficile détermination de l'objet.....	17
A) Distinction entre idée et expression originale.....	18
1) Une exclusion de principe.	18
2) Une protection indirecte.	19
B) La mise à disposition.	19
1) Ce qui est remis.	19
2) L'intelligence partagée.	20
II) L'inadaptation du droit.	20
A) La décompilation	20
1) Une exception en droit commun.....	20
2) Un principe pour les logiciels libres.	22
B) Un rejet du droit d'auteur.....	22
CHAPITRE II : EN TANT QUE PRODUIT INDUSTRIEL.	24
<i>Section 1 : Une exclusion de principe.</i>	<i>24</i>
I) Fondement du rejet.....	24
A) Appréciation malaisée du caractère technique.....	25
1) L'existence d'un problème technique.....	25
2) L'absence d'une solution technique.	25
B) La contribution à l'état de la technique.....	26
1) Le caractère inventif.	26
2) Le caractère nouveau.	27
II) Un péril pesant sur la communauté du libre.....	27
A) Une menace économique.	28
B) Une menace juridique.	28
1) La peur du tout brevetable.	28

2) Les risques d'une multiplication du contentieux.	29
Section 2 : La remise en cause du principe.	30
I) L'alternative d'une protection per se.....	30
A) De manière indirecte.	31
1) En tant que composant d'un procédé technique.	31
2) Par l'adaptation des critères de brevetabilité.	31
B) Par un revirement de la logique juridique.	32
II) Etendue de la protection du brevet de logiciel.	32
A) Appréhension de l'objet.	32
1) Par les revendications.	32
2) Par sa délimitation avec les méthodes et les algorithmes.	33
B) Vers un cumul des protections ?	34
1) Un choix entre coexistence et cumul.	34
2) En matière de contrefaçon.	34
TITRE II : UNE PROTECTION AD HOC.....	36
CHAPITRE I : LE RECOURS AU CONTRAT.....	37
Section 1 : Typologie des diverses licences.....	37
I) Les licences ouvertes.....	38
A) Les licences de domaine public.	38
B) Les licences semi-libres	38
II) Les licences de type GNU.....	39
Section 2 : Leur appréhension par le droit commun des contrats.	40
I) L'encadrement d'une communauté.	40
A) Un lien contractuel unioniste.	40
B) La formation du contrat.	41
II) L'exercice de la liberté à travers le contrat.....	42
A) Droits et obligations du licencié.	42
1) Le fonds commun.	42
2) Les effets du contrat.	42
B) Les limites.	44
1) Un manque de conformité au droit.	44
2) Le lien viral.	45
CHAPITRE II : LA CATHEDRALE ET LE BAZAR : L'APPROPRIATION DE	
L'INFORMATION.....	47
Section 1 : La cathédrale, l'idée du tout appropriable.	48
I) Les différents modes de réservation.	48
A) Le prix du savoir	48
B) La protection du savoir	48
II) L'applicabilité aux logiciels libres.....	49
Section 2 : Le bazar, la circulation du savoir.	50
I) Principe fondateur de la communauté du libre.....	50
II) L'applicabilité du modèle libre à d'autres créations.....	51
A) Une réalité hors de l'univers numérique.	51
B) Mais une illusion pratique.	52

ANNEXES

Sommaire : 1) Licence GNU.....	52
2) Exemples de logiciels libres.....	56

Annexe 1 : Licence GNU



GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- **a)** You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- **b)** You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- **c)** If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- **a)** Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- **b)** Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- **c)** Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients'

exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Annexe 2 : Exemples de logiciels libres.



- Les systèmes d'exploitation : Linux, FreeBSD, NetBSD, OpenBSD et Hurd



- Les logiciels de traitement d'images Gimp, Gsview ou Blender
- Les environnements de programmation Gcc, G++, Perl, Php, Python, Scheme, Caml, Tcl/Tk, MesaGL...



- Les bases de données relationnelles MySQL, POSTGRES
- L'ensemble des outils GNU tels que Gawk, Gmake...
- Le serveur Web Apache, le serveur de newsgroups Inn. Samba



BIBLIOGRAPHIE.

Ouvrages généraux :

- André Bertrand, "Le Droit d'Auteur et les Droits Voisins", 2^{ème} édition, 1999, Dalloz.
- Claude Colombet, "Propriété littéraire et artistique et droits voisins", 9^{ème} édition, 1999, Dalloz.
- André Lucas, "Droit d'Auteur et Numérique", 2^{ème} édition, 1998, Litec.
- J. Carbonnier, "Droit Civil", 12^{ème} édition, Thémis T.4
- Lamy droit de l'informatique et des réseaux, édition 2001
- André Lucas, Jean Devèze et Jean Frayssinet, "Droit de l'informatique et de l'internet", 2001, PUF Thémis droit privé
- Frédéric Pollaud-Dulian, "droit de la propriété industrielle", 1999, Montchrestien
- André et Henri-Jacques Lucas, "Traité de la propriété littéraire et artistique", 2^{ème} édition, Litec

Sites sur le sujet :

<http://www.aful.org>

<http://www.gnu.org>

<http://www.freepatents.org>

<http://www.april.org>

<http://www.alcove.com>

<http://www.fsfeurope.org>

<http://www.librelogiciel.com>

<http://www.opensource.org>

<http://www.redhat.com>

<http://emoglen.law.columbia.edu>

<http://www.droit-ntic.com>

<http://artlibre.org>

REMERCIEMENTS A... :

- Monsieur Sirinelli
- Monsieur Vivant
- Madame Clement Fontaine
- Mademoiselle Modat
- Monsieur et Madame Spiers
- Monsieur R.Linsolas
- Monsieur Vannier
- Monsieur Macrez
- Aux membres du DESS

...pour leur aide précieuse.