

La protection juridique du logiciel en droit français est-elle suffisante ?

par Raphael Marichez ([Home Page](#))

Date de publication : 15/01/2008

Dernière mise à jour :

Mémoire réalisé dans le cadre du cours MJSEC à l'ENST Paris en mars 2006.

I - La protection juridique du logiciel en droit français

I-A - Les besoins en terme de protection du logiciel.

I-A-1 - La reconnaissance de l'auteur

I-A-2 - La protection de l'investissement économique

I-A-3 - La protection des droits de l'utilisateur et la libre concurrence

II - Doit-on faire évoluer la législation ?

II-A - Insuffisances actuelles de la loi

II-B - Un brevet logiciel, dans le respect des protections dont bénéficient le consommateur

II-B-1 - Autorisation des brevets sur les méthodes et les algorithmes

II-B-2 - Restriction aux brevets sur les programmes et non sur les méthodes

I - La protection juridique du logiciel en droit français

L'économie actuelle se développe principalement autour de l'outil informatique. C'est donc tout naturellement que voient le jour de plus en plus de logiciels. Dans un monde idéal où chacun respecterait une éthique lui interdisant de profiter sans mesure des créations d'un autre, et surtout à ses dépens, l'auteur d'un logiciel n'aurait pas d'autres soucis que de se concentrer sur les fonctionnalités même de son logiciel, ainsi que sur sa promotion.

Cette situation est malheureusement très instable : le moindre écart à cette règle # même une seule personne qui profiterait du système # entraîne la chute de ce système. Il est donc nécessaire pour le bon développement de notre économie, et pour favoriser la création, de fixer ces «règles morales» sous la forme d'une loi. Aujourd'hui, pour ce qui est de la protection du logiciel, il s'agit principalement du Code de la Propriété Intellectuelle. Il a vu le jour au XVIIIe siècle sous l'impulsion de Beaumarchais qui souhaitait avoir une meilleure reconnaissance de sa qualité d'auteur de pièces de théâtre, reconnaissance en particulier dans le domaine financier afin de lui permettre de vivre de ses créations littéraires.

Depuis ses débuts, cette loi a régulièrement évolué afin de prendre en compte l'évolution de notre société, en particulier l'évolution de la technique. Dans le domaine des logiciels, celui qui nous intéresse, les industriels parlent de plus en plus de *brevets logiciels*, notion qui a un sens aux États-Unis, mais pas en France. En effet, contrairement à nos amis d'Outre-Atlantique, la France et les autres pays de l'Union Européenne n'autorisent pas déposition d'un brevet sur un logiciel, un algorithme, même s'ils sont novateurs.

Au vu de cet important lobby de l'industrie du logiciel en faveur des brevets logiciels, nous nous devons de nous interroger sur la question sous-jacente : la protection juridique du logiciel en droit français est-elle suffisante ? Afin d'apporter notre réflexion sur ce sujet, nous commencerons par exprimer les besoins des programmeurs en terme de protection légale sur leur logiciel, ainsi, bien sûr, que ceux des utilisateurs qui ne doivent pas non plus être lésés. Nous étudierons ensuite l'adéquation de la législation actuelle au vu de ces besoins, et en particulier ses manques.

I-A - Les besoins en terme de protection du logiciel.

I-A-1 - La reconnaissance de l'auteur

Un logiciel est le fruit de l'expression d'une idée. Tout comme un écrit, une photo, une musique, il est composé d'un fond et d'une forme. Le fond correspond au contenu lui-même, au but dans lequel il a été créé. Dans le cas d'un logiciel, il s'agit de toute la phase de sa conception : les idées de fonctionnalités, les algorithmes, son organisation générale. Ce fond étant complètement immatériel, il a besoin d'être concrètement exprimé. C'est là que joue le savoir-faire du programmeur, afin de donner sa forme au logiciel, tout comme un écrivain mettrait en phrases ses idées, et choisissant les mots permettant d'atteindre au mieux son but.

Cette description du cheminement de la création d'un logiciel montre que ce dernier est complètement le fruit d'une création intellectuelle, raison pour laquelle la loi actuelle classe les logiciels dans les «œuvres de l'esprit». Étant le produit de l'imagination d'un homme, il est important que le logiciel en résultant reste attaché à cette personne, par exemple par l'intermédiaire de son nom. De même, une autre personne ne doit pas pouvoir reprendre cette création sous nom, ce qui diffuserait à tort l'information qu'elle en est aussi l'auteur, et donc limiterait la paternité effective de l'auteur réel, telle que la percevrait alors l'opinion publique.

Cependant, si le logiciel est rattaché à son auteur, l'image véhiculée par le logiciel se répercute sur celle de l'auteur. Il est donc important que l'image transparaissant du logiciel soit sous le contrôle de l'auteur : soit parce qu'il s'agit de ce qu'il a personnellement écrit, soit parce qu'il a donné son accord pour effectuer une modification. L'accord peut toutefois être implicite lorsque les modifications sont mineures et ne peuvent vraisemblablement pas impacter l'image que donne le logiciel : l'important étant surtout de respecter l'esprit originel donné par l'auteur.

Cette image dépend aussi de la qualité du logiciel. Étant donné que cette qualité croît généralement avec le temps mis à le réaliser, l'auteur doit pouvoir lui-même décider à quel moment son travail est terminé, c'est-à-dire à quelle date son logiciel pourra être diffusé ; une diffusion prématurée sans son accord d'une version non finalisée ne pouvant que lui porter préjudice.

Pour les mêmes raisons de paternité de l'idée originale et de respect de l'image de l'auteur, celui-ci peut souhaiter empêcher la création d'un logiciel similaire au point que les utilisateurs aient du mal à faire la différence entre les deux logiciels, comme le cas d'OpenOffice reprenant quasiment à l'identique les interfaces et les fonctionnalités de Microsoft Office. La duplication exacte ne fait pas partie de ce cas, car elle ne peut pas se différencier de l'original. Elle ne peut porter préjudice à l'auteur que du point de vue du profit qu'il peut tirer de ce logiciel, comme nous le verrons dans la partie suivante.

I-A-2 - La protection de l'investissement économique

Notre société actuelle est profondément ancrée autour de l'argent. Un auteur, comme toute autre personne, a besoin d'argent pour se nourrir, se loger. L'écriture de logiciel est une activité qui demande souvent de nombreuses ressources en temps, il n'est donc pas toujours possible d'exercer une autre activité rémunératrice en parallèle.

L'équilibre entre l'intérêt financier du créateur et l'intérêt de l'utilisation du produit par le public est nécessaire à la bonne marche de l'économie. L'industrie pharmaceutique est un exemple flagrant de domaine dans lequel il est primordial de permettre un accès aisé des utilisateurs aux médicaments. Seulement, la protection grâce aux brevets protège les investissements financiers qui permettent le développement de la recherche, ce qui ne va pas sans créer de conflits avec les pays en voie de développement, très demandeurs de médicaments, comme lors du procès de Prétoria en 2001, sur les brevets sur les anti-rétroviraux¹. Les uns font appel à la nécessité pour le plus grand nombre d'accéder à des médicaments à bas prix, les autres mettent en avant la protection de leurs investissements financiers et donc leur rentabilité. En matière de logiciel aussi, se pose la problématique du financement des activités de recherche et de production, préalables aux premières ventes, et donc diffusions, du produit fini.

De cet aspect très pratique, ressort un besoin plus général : permettre à l'auteur de choisir les modalités de diffusion. En effet, même si l'auteur souhaite diffuser son logiciel gratuitement, il est légitime qu'il puisse interdire à d'autres personnes d'en faire du profit, ou simplement garantir que même en le redistribuant à titre onéreux, les clients de ces autres personnes aient accès au code source, et puissent le modifier librement, comme le prévoit par exemple la licence GPL²

Ainsi, il est important qu'un auteur, ou le distributeur de logiciels qu'il a choisi, puisse contrôler la manière dont les copies sont réalisées. Ce contrôle s'effectue grâce à un contrat avec l'utilisateur, souvent appelé licence d'utilisation. La revendication principale d'un distributeur de logiciel est de réclamer qu'un logiciel acheté sous la forme d'un CD-ROM, ou *via* un téléchargement en ligne, ne puisse être installé que sur une unique machine à un instant donné. Il s'agit là de mimer ce qu'il se passe lors de la vente d'un bien matériel, comme un aspirateur : acheter un CD-ROM d'un logiciel puis l'installer sur plusieurs stations reviendrait alors à acheter un aspirateur et le multiplier pour l'utiliser en même temps dans le salon et la chambre. De ce point de vue, il est légitime qu'un distributeur de logiciels puisse imposer qu'il n'y ait, lorsqu'il vend n licences d'utilisations d'un logiciel, que n copies installées de ce logiciel, ou n instances du logiciel lancées en même temps.

Jusque là, nous ne nous soucions que des copies strictes, identiques bits à bits, du logiciel. Ce n'est cependant pas le seul souci d'un auteur de logiciel, nous reviendrons donc sur le problème de la réécriture complète du logiciel par une autre personne. Contrairement au point de vue de la partie précédente qui était la potentielle dévalorisation de l'image du logiciel originel à cause de la mise en circulation d'une copie de mauvaise qualité, nous nous intéresserons ici aux copies ayant des fonctionnalités similaires et pouvant impacter le profit effectué par l'auteur du logiciel original. Ici, le mot profit ne signifie pas uniquement profit financier, mais aussi toutes autres sortes de gains, et en particulier, la renommée ou la diffusion d'une philosophie, comme c'est le cas pour le logiciel libre. Idéalement, un auteur pourrait souhaiter qu'aucun logiciel ayant des fonctionnalités similaires ne puisse être diffusé. L'existence potentielle d'un tel

monopole drastique révèle le besoin de limiter le pouvoir de protection d'un auteur sur son logiciel, afin de ne pas léser l'utilisateur au profit de l'auteur, ce que nous allons aborder dans la partie suivante.

*

Florent Latrive, Du bon usage de la piraterie, Introduction. Exils éditeur, 2004

*

Licence GPL : <http://www.gnu.org/copyleft/gpl.html> (anglais) <http://fsffrance.org/gpl/gpl-fr.fr.html> (français)

I-A-3 - La protection des droits de l'utilisateur et la libre concurrence

Nous venons de recenser certains besoins de protection du logiciel, mais il ne s'agit là que du point de vue de l'auteur. En effet, un logiciel est fait pour être utilisé, il faut donc au moins garantir à l'utilisateur qu'il aura le droit de s'en servir !

La liberté d'utilisation et la liberté de la concurrence sont intimement liées. La question de la liberté d'utilisation d'un algorithme ou d'une technique novatrice concerne moins l'utilisateur final que le programmeur d'un futur produit, qu'il réponde au même besoin ou non, et qui ferait appel à des techniques très récemment développées. Du point de vue de la liberté d'entreprendre, il est difficile de se refuser à utiliser un procédé reconnu et performant sous prétexte qu'il est breveté, alors qu'il répond au besoin du programmeur, et y répond même d'autant mieux que le programmeur ne connaît pas de solution plus efficace et rapide.

Lors de cette considération, il s'agit non de recopier des lignes de codes déjà existantes, mais bien de construire un logiciel tirant partie d'une idée, d'un principe ou d'une méthode, déjà utilisé et revendiqué par une personne morale. Ce principe peut prendre la forme d'un parcours algorithmique, d'une structure de données efficace relativement à tel ou tel usage, d'une méthode permettant d'imbriquer un programme dans un module plus grand, ou encore d'un comportement proposé à l'utilisateur, comme le clic-droit pour faire apparaître un menu contextuel# pour reprendre des exemples qui ont été discutés Outre-Atlantique. Nous sommes donc loin de la considération de code source *ouvert* ou *fermé*, puisque les entreprises peuvent souhaiter protéger certains comportements du logiciel, visibles à l'utilisateur.

Dans un objectif de diffusion du savoir et des technologies, les principes et méthodes de base doivent donc pouvoir être utilisés par la concurrence pour construire leurs logiciels propres. La question est plus délicate s'il s'agit de créer un logiciel rendant un service identique, avec les mêmes innovations technologiques # qui ne sont donc plus des innovations #, et vendu à moindre coût.

La copie conforme d'un logiciel, quant à elle, est actuellement encadrée par les licences d'utilisation. Celles-ci définissent par exemple combien de copies un utilisateur peut effectuer pour son usage personnel, éventuellement pour des sauvegardes, mais aussi combien d'instances du logiciel peut utiliser simultanément une entreprise sur les postes de travail de ses employés. Comme dit précédemment, la loi doit permettre au propriétaire du logiciel de contrôler l'usage qui en est fait, mais sans empêcher le propriétaire d'en autoriser la copie pour un certain usage, voire la diffusion en tant que logiciel libre, s'il le souhaite. Dans ce domaine spécifique du logiciel libre, le problème de l'exclusivité d'une invention peut entrer en conflit avec la liberté de modifier, d'améliorer le logiciel, ou de s'en inspirer pour d'autres réalisations, sous certaines conditions définies dans les licences.

II - Doit-on faire évoluer la législation ?

II-A - Insuffisances actuelles de la loi

Maintenant que nous avons dressé l'ensemble des besoins qu'ont les différentes parties # auteurs, éditeurs et utilisateurs de logiciels #, se pose tout naturellement la question de l'adéquation de la législation actuelle sur la protection du logiciel. La principale loi qui traite de ce sujet est le Code de la Propriété Intellectuelle, mais ce n'est pas la seule : les licences d'utilisations, elles, sont régies par les textes du code civil et du code de commerce relatifs aux contrats.

La loi française classe le logiciel dans la catégorie des « oeuvres de l'esprit », ce qui permet d'appliquer à la protection des logiciels l'ensemble des lois relatives aux droits d'auteurs. Cela permet en pratique de garantir les droits moraux³ # la paternité, le choix de la date de première divulgation, le respect de son nom et de son oeuvre #, et les droits patrimoniaux⁴ # contrôle de la reproduction. Dans le cas particulier du logiciel, la copie privée est expressément interdite, il n'est donc pas possible d'invoquer l'exception de copie privée pour installer le logiciel acheté sur plusieurs ordinateurs, l'installation s'apparentant à une copie. Les seules copies autorisées sont celles destinées à des fins exclusivement de sauvegarde, pour palier la destruction potentielle du support d'origine contenant le logiciel acheté. On se rapproche donc beaucoup de la situation existante pour les biens matériels, avec de plus un avantage pour le client, inerrant au monde numérique : la garantie, s'il prend les dispositions nécessaires via la création de sauvegardes, de ne jamais voir son bien acheté se dégrader.

Pour revenir plus précisément sur la distribution d'un logiciel par un auteur, ou par l'éditeur qu'il a choisi, regardons les possibilités qui s'ouvrent à eux. La personne détentrice des droits patrimoniaux a le choix des modalités de diffusion de supports contenant le logiciel, et peut donc mettre en place un contrat de vente avec son client. Ce contrat prend généralement le nom de licence d'utilisation, mais reste bien un contrat régi par le code civil qui en particulier rend non applicables les éventuelles clauses abusives. La protection du logiciel par la licence se limite donc là où elle commencerait à nuire au bon usage du logiciel.

La législation en vigueur en France est donc déjà très complète au sujet de la protection des logiciels. Néanmoins certains points semblent avoir été partiellement oubliés. Quel éditeur de logiciel ne met pas de clause d'exonération de responsabilité des dommages qui pourraient être dus à l'utilisation de ses logiciels ? Cette pratique semble autorisée par loi étant donné qu'elle est généralisée et qu'aucune entreprise n'a encore été inquiétée par la justice à ce sujet. Cependant, il est clair qu'il s'agit d'une atteinte profonde au droit des utilisateurs. En effet, quelle est la différence entre un logiciel qui pourrait par erreur détruire le contenu d'un disque et donc des musiques achetées sur Internet, et un aspirateur qui, à cause d'un défaut de conception, exploserait et, pour rester dans le domaine de la musique, détruirait l'ensemble de la CD-thèque matérielle de l'utilisateur ? Dans le cas d'un outil matériel, on pourrait se retourner contre le fabricant, mais pas dans le cas d'un outil immatériel ? Les mêmes réflexions s'appliquent à la garantie légale dans le cas d'existence de vices cachés. Ce problème est en partie résolu par une recommandation de la commission des clauses abusives⁵, mais son application tarde à être réellement effective.

*

Articles L121-1 et suivants du Code de la Propriété Intellectuelle

*

Articles L122-1 et suivants du Code de la Propriété Intellectuelle

*

Recommandation de la Commission des Clauses Abusives n°95-02 adopté le 7 avril 1995

Pour retourner du côté des éditeurs, ceux-ci ont l'impression de ne pas être assez protégés par la loi sur les droits d'auteur qui laisse explicitement libres les idées et concepts. En effet, rien n'interdit à un autre programmeur de reprendre l'ensemble des idées originales et des principes utilisés par un autre logiciel, afin de réécrire un logiciel similaire tant qu'il apporte un minimum de travail intellectuel et donne à son propre programme une identité propre, différente de celle du programme dont il s'est inspiré. Les industriels français souhaiteraient pouvoir protéger leur algorithmes ou idées innovantes, ce souhait étant à l'origine de la création d'un lobby pro-brevets logiciels.

Avant de discuter des potentiels intérêts des brevets logiciels, remarquons qu'il est déjà possible de breveter indirectement un logiciel⁶. Le terme "indirectement" signifie qu'il est possible de breveter un dispositif matériel complet formant un tout et contenant une composante logicielle.

*

Article L611-10 alinéa 3 du Code de la Propriété Industrielle

II-B - Un brevet logiciel, dans le respect des protections dont bénéficient le consommateur

Rappelons tout d'abord l'article du Code de la Propriété Intellectuelle relatif aux inventions brevetables, qui interdit les brevets logiciels (article L611-10 alinéa 2) :

2. Ne sont pas considérées comme des inventions au sens du premier alinéa du présent article notamment :

a) Les découvertes ainsi que les théories scientifiques et les méthodes mathématiques ;

b) Les créations esthétiques ;

c) Les plans, principes et méthodes dans l'exercice d'activités intellectuelles, en matière de jeu ou dans le domaine des activités économiques, ainsi que les programmes d'ordinateurs ; (#)#

Un amendement de cette partie législative, qui autoriserait les brevets sur les programmes d'ordinateurs, en supprimant la mention *ainsi que les programmes d'ordinateurs* de l'énumération des inventions non brevetables, serait délicat dans son champ d'application. En particulier, les principes, méthodes et algorithmes ne seraient toujours pas brevetables en l'état, en application des alinéas 2-a et 2-c de l'article sus-cité : *ne sont pas considérées comme des inventions (#) notamment : a) Les découvertes ainsi que les théories scientifiques et les méthodes mathématiques ; (#) c) Les plans, principes et méthodes dans l'exercice d'activités intellectuelles*. En effet, un algorithme n'est pas lié au domaine informatique, et peut être transposé en une méthode mathématique. D'ailleurs, de nombreux algorithmes massivement utilisés dans les logiciels sont directement issus de la recherche mathématique # arithmétique, théorie des nombres ou des ensembles, cryptographie#

Une modification plus profonde de cet article, en particulier de ses alinéas 2-a et 2-c, nous amènerait alors à la situation que connaissent les États-Unis, où peuvent être brevetés des concepts très larges tels que le *plug-in*, le *clic-droit* de souris, le format de description de données XML, etc.

II-B-1 - Autorisation des brevets sur les méthodes et les algorithmes

Dans ce cas, sans préjuger d'autres conséquences dans les disciplines autres que le logiciel, il serait impossible d'utiliser ces méthodes brevetées dans les logiciels libres. En effet, si une telle méthode brevetée se trouvait être utilisée dans un logiciel libre, l'exploitation libre de ce logiciel entrerait en contradiction avec le brevet. Les utilisateurs, pour des raisons professionnelles ou personnelles, de systèmes à base de logiciels libres, seraient donc considérablement lésés.

Cela pose également le problème d'inter-opérabilité des structures de documents : un document utilisant un format breveté ne pourrait être visionné, ni modifié, que par un logiciel autorisé par le titulaire du brevet. Cela semble très prometteur d'un point de vue lucratif pour le titulaire du brevet, mais se posent alors des questions de libre concurrence, pour les autres éditeurs de logiciels, et également d'utilité, pour les utilisateurs. La situation de monopole ou d'oligopole qui serait ainsi engendrée, le serait évidemment au déficit de l'utilisateur. Ceci est encore plus vrai dans le cas d'une faillite de l'éditeur ayant souscrit un contrat de maintenance # ou ayant l'habitude de maintenir son logiciel # avec ses clients. Néanmoins, l'article L122-6-1 alinéa I du Code de la Propriété Intellectuelle peut sauver ici l'utilisateur :

*I. Les actes prévus aux 1^o et 2^o de l'article L. 122-6⁷ ne sont pas soumis à l'autorisation de l'auteur lorsqu'ils sont nécessaires pour permettre l'utilisation du logiciel, conformément à sa destination, par la personne ayant le droit de l'utiliser, **y compris pour corriger des erreurs.***

Toutefois, l'auteur est habilité à se réserver par contrat le droit de corriger les erreurs et de déterminer les modalités particulières auxquelles seront soumis les actes prévus aux 1^o et 2^o de l'article L. 122-6, nécessaires pour permettre l'utilisation du logiciel, conformément à sa destination, par la personne ayant le droit de l'utiliser.

L'utilisateur a donc la possibilité de désassembler et modifier le logiciel à des fins de corrections d'erreurs, de bogues. Le brevet logiciel ne devrait donc pas pouvoir, en aucun cas, entraver ce droit.

De plus, le concept de l'interopérabilité entre logiciels figure déjà dans la législation française sous l'article L122-6-1 alinéa IV du code de la propriété intellectuelle :

IV. La reproduction du code du logiciel ou la traduction de la forme de ce code n'est pas soumise à l'autorisation de l'auteur lorsque la reproduction ou la traduction au sens du 1^o ou du 2^o de l'article L. 122-6 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^o 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^o Les informations nécessaires à l'interopérabilité n'ont pas déjà été rendues facilement et rapidement accessibles aux personnes mentionnées au 1^o ci-dessus ;

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

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

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

3^o 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.

*

NdA: la reproduction, la traduction, l'adaptation, l'arrangement, la modification#

Est entendu que la traduction du logiciel à des fins autres que l'interopérabilité relève de la contrefaçon.

Dès lors, étant donné que toute personne # sous les trois conditions visées par l'article # peut décompiler le code d'un logiciel d'origine afin de procéder à l'interopérabilité, par exemple, de structures de données entre un logiciel et un

autre⁸, cette personne peut être amenée à exploiter, dans son propre logiciel, une méthode ou un algorithme breveté, à des fins d'inter-opérabilité avec le logiciel d'origine. Comment concilier cela avec les restrictions sur l'exploitation de la méthode ou l'algorithme dues au brevet ?

D'autre part, toujours dans l'hypothèse où les brevets sur les méthodes et les algorithmes seraient autorisés, l'application de la loi à un auteur de logiciel dérivé de cette méthode ou de cet algorithme serait problématique. En effet, le caractère breveté d'une invention n'interdit non pas sa divulgation, mais bien son exploitation par un tiers # sans autorisation du titulaire du brevet, bien entendu. L'article L613-15 du Code de la Propriété Intellectuelle définit les conditions à l'exploitation d'une invention brevetée issue d'un brevet antérieur.

Le titulaire d'un brevet portant atteinte à un brevet antérieur ne peut exploiter son brevet sans l'autorisation du titulaire du brevet antérieur ; ledit titulaire ne peut exploiter le brevet postérieur sans l'autorisation du titulaire du brevet postérieur.

Lorsque le titulaire d'un brevet ne peut l'exploiter sans porter atteinte à un brevet antérieur dont un tiers est titulaire, le tribunal de grande instance peut lui accorder une licence d'exploitation du brevet antérieur dans la mesure nécessaire à l'exploitation du brevet dont il est titulaire et pour autant que cette invention constitue à l'égard du brevet antérieur un progrès technique important et présente un intérêt économique considérable. (#)

Une personne tierce peut donc prendre connaissance de la méthode novatrice qui fait l'objet du brevet dit *antérieur*, créer un logiciel s'inspirant de cette méthode, et l'exploiter # comprendre : le diffuser ou le vendre. La nature fermée du logiciel dont les sources ne sont pas nécessairement disponibles rend difficile l'appréciation du terme *s'inspirer de cette méthode*. Le logiciel représente bien un progrès notable par rapport à la simple méthode initialement brevetée, nous sommes donc bien, sur ce point, dans le cadre de l'article L613-15. Or le programmeur fera valoir ses droits à l'exploitation de son oeuvre, et demandera peut-être à breveter son logiciel. Le titulaire du brevet antérieur pourrait accuser le programmeur d'avoir utilisé sa méthode novatrice à des fins d'exploitation, entrant ainsi en conflit sur les termes de l'exploitation du logiciel. Le titulaire du brevet antérieur avancera la présomption selon laquelle le nouvel exploitant utilise la méthode brevetée, mais sans pouvoir nécessairement le prouver. L'article renvoie ce type de litige devant le tribunal de grande instance. Nul doute qu'une modification législative en ces termes entraînerait de nombreux recours devant les tribunaux, du moins si le cadre de l'article L613-15 n'est pas plus strictement redéfini.

*

Marie Barel et Frédéric Raynal, Vulnérabilités logicielles et droit : responsabilité, recherche et divulgation, in MISC n°23 (janvier-février 2006), pp.16-23, Diamond Editions.

II-B-2 - Restriction aux brevets sur les programmes et non sur les méthodes

En conséquence, limitons-nous uniquement à l'hypothèse de la suppression stricte de la mention *ainsi que les programmes d'ordinateurs* de l'article L611-10 du CPI, en conservant pour acquis l'interdiction de brevet sur un algorithme ou sur une méthode. Ceci permettrait alors le dépôt d'un brevet sur une réalisation logicielle, novatrice, susceptible d'application industrielle, c'est-à-dire offrant directement un service à l'utilisateur. En ce sens, la copie *bit à bit* non explicitement autorisée d'un logiciel, suivie de son exploitation, serait illégale : mais elle l'est déjà en vertu des lois sur la contrefaçon, et souvent de la licence d'exploitation. Sur ce point donc, la valeur ajoutée de cette modification est nulle. Notons par ailleurs que dans ce cas, un logiciel libre ne peut donc être soumis à un brevet, puisque le logiciel libre doit pouvoir être exécuté librement, à discrétion.

Cependant, se pose la question de la réalisation *ex nihilo* d'un logiciel aux fonctionnalités équivalentes, avec des moyens de développements entièrement distincts. Là encore, nous entrons dans des difficultés d'appréciations. Il est tentant, et utile du point de vue du consommateur, de recoder depuis zéro un logiciel remplissant à peu près les mêmes fonctions qu'un logiciel antérieur propriétaire et payant. A l'excès, nous avons vu que cela peut aller à un manque à gagner certain pour l'éditeur du logiciel antérieur. A l'inverse, avec une protection excessivement forte par brevet sur les fonctionnalités apportées par le logiciel antérieur, s'il est interdit de développer et d'exploiter un nouveau

logiciel répondant au même besoin, nous entrons dans la situation de monopole, sans parler des technologies sous-jacentes # format de stockage des données, etc. #, que nous avons traitées au paragraphe précédent.

Ce type d'excès est habituellement modéré par l'alinéa 1 de l'article L611-2 du CPI, protégeant les inventions *pour une durée de vingt ans*. Cette durée permet, dans les applications industrielles conventionnelles, la rentabilisation des investissements ayant abouti à l'innovation, sans entraver le progrès technologique à long terme. Cependant, en matière de logiciel, les investissements sont bien moindres aussi bien en termes financiers qu'en termes de durée, et les manques à gagner # ou les bénéfices # peuvent être considérables durant les toutes premières années d'exploitation du logiciel. Nous sommes donc très loin des vingt années spécifiées par la loi, mais nous constatons la difficulté de protéger suffisamment l'inventeur durant les premiers mois d'exploitation de son innovation.

Le juste compromis législatif peut s'avérer difficile à trouver, et pour deux raisons essentielles. D'une part, l'appréciation de la proximité de deux logiciels pourra souvent mener à débat. D'autre part, les recours devant le tribunal de grande instance, prévus notamment par l'article L613-15 du CPI dans le cadre de l'exploitation d'un brevet portant atteinte à un brevet antérieur, et par l'article L615-19 du CPI dans les cas de contrefaçons, risquent d'être nombreux, du moins tant qu'une jurisprudence n'est pas clairement établie.

Enfin, le logiciel, outre sa capacité à être développé, exploité et distribué très rapidement en regard des applications industrielles classiques, possède aussi la capacité à traverser les frontières très facilement. Ainsi, une protection par brevet contre l'exploitation par un tiers d'un logiciel sur le territoire national, n'empêchant donc pas la réalisation et l'exploitation d'un logiciel similaire à l'étranger, pas plus que son importation sur le territoire national # sauf cas particuliers #, ne peut s'appliquer contre l'exploitant résident à l'étranger. Le consommateur quant à lui peut utiliser ce logiciel similaire, bien entendu tant qu'il n'en effectue pas une exploitation sur le territoire national portant atteinte au brevet.

Les logiciels, bien qu'étant déjà protégés par le code de la propriété intellectuelle en qualité d'oeuvre de l'esprit, souffrent ainsi d'un manque de protection lorsqu'il s'agit d'innovations que l'on pourrait qualifier d'industrielles, dès lors qu'elles mettent en jeu de la recherche, de l'exploitation, du commerce et des flux financiers. À ce titre, certains pays ont amendé leur législation pour y inclure le droit de breveter un logiciel, voire un algorithme, une méthode, un format, etc. D'autre part, des lobbies industriels cherchent à s'affranchir de ces différences nationales, grâce notamment à l'éventuelle reconnaissance mutuelle des brevets sur le territoire de la Communauté Européenne⁹, en cours de discussion actuellement¹⁰.

Néanmoins, ces éventuels brevets logiciels, de par la nature même du secteur informatique, ne peuvent pas être considérés sur le même plan que les brevets d'innovation traditionnels. Une modification du droit français amenant à légaliser le dépôt de brevets sur les logiciels n'irait pas sans certains excès, s'il n'y a pas en amont une réflexion précise sur les modifications à apporter, ou du moins des dispositions encadrant strictement ce type de brevets, et permettant de concilier les intérêts des créateurs et ceux du public.

*

Brevet communautaire, Approche politique commune. Note 7159/03 du 7 mars 2003 du Conseil de l'Union européenne, dossier inter-institutionnel 2000/0177. <http://register.consilium.eu.int/pdf/fr/03/st07/st07159fr03.pdf>

*

Consultation de la Commission européenne sur le brevet communautaire (mars 2006) : <http://www.ffii.fr/consultation-brevet-communautaire-ffii-france>

