

Extrait du FFII.FR

<http://ffii.fr/Logiciel-et-liberte-d-expression>

# Logiciel et liberté d'expression

- Brevets logiciels - Analyses générales -

Date de mise en ligne : lundi 7 février 2005

---

**FFII.FR**

---

- 1. Les logiciels sont des écrits. En tant que tels, ils doivent être soumis au régime du droit d'auteur (dans son acceptation étroite) ou à la protection par le secret industriel, et non à la loi sur le brevet. En tant qu'écrits, les programmes doivent être protégés contre toute tentative du gouvernement visant à restreindre la liberté d'expression. Les arguments légaux ou politiques, même bien intentionnés n'en sont pas moins dans l'erreur, s'ils engendrent censeurs et censure ; les désigner par les termes « examinateurs de brevets » et « examen de brevets » ne changera rien au fait.**

**Une telle censure, une telle entrave à la liberté d'expression des créateurs de logiciels est un anathème dans une société libre, et une violation du Premier amendement. Les brevets logiciels sont une violation majeure du droit à l'expression des programmeurs malgré que ce fait soit encore peu connu ; c'est peut-être en partie parce que la plupart des juristes, juges et politiciens sont encore insuffisamment formés à la problématique des ordinateurs pour réaliser qu'écrire un logiciel est réellement une forme d'écriture, guère plus ésotérique que la composition musicale, les mathématiques, la rédaction d'articles scientifiques ou, en l'occurrence, de textes de loi. Toutes ces formes d'expression, y compris l'écriture de programmes, méritent une protection totale selon le Premier amendement.**
- 2. La planification centralisée ou la soumission à un régime de licence des bonnes idées ne fonctionnera pas pour le logiciel. Comme serait vouée à l'échec toute tentative de centraliser ou classer dans un office des brevets les écrits littéraires, musicaux, ou scientifiques selon un critère d'originalité (ou de « non-évidence »), toute tentative de centraliser l'information concernant les logiciels innovants échouera également. Personne ne peut connaître tous les logiciels relatifs à un domaine, de même que personne ne peut connaître l'ensemble de ce qui a été écrit sur un sujet particulier, et pour les mêmes raisons. Les innovations récentes dans le domaine des logiciels engendreront la production de programmes à un taux toujours croissant. En conséquence, la charge de travail de l'office des brevets, nécessaire à la gestion centralisée de l'innovation deviendra de plus en plus onéreuse, créant par là des barrières inutiles et coûteuses au progrès informatique.**
- 3. L'évolution des idées grâce à l'apprentissage par l'erreur est entravée par la brevetabilité des écrits. Plutôt que de permettre au gouvernement de restreindre les diverses expressions d'une même idée, au moyen de brevets ou autrement, une politique publique devrait refléter le fait que plus une idée est importante, plus il faut encourager les variations les plus libres possibles de cette expression, afin de parvenir rapidement à perfectionner l'idée. L'intense concurrence de ces dernières années dans le secteur du logiciel, et l'amélioration rapide des pratiques de développement des programmes, que le marché libre a favorisée jusqu'à présent, valident fortement cette théorie.**

*Traduction française de Gilles Sadowski*

## Les logiciels sont des écrits

Quiconque a écrit un programme et un essai littéraire, sait combien ces exercices sont similaires. Tous deux requièrent l'utilisation de toute notre habileté, de tout notre savoir. Tous deux nécessitent continuellement inventivité et créativité. Tous deux imposent des révisions constantes. Tous deux évoluent dans le temps, en fonction de l'élargissement du champ de nos connaissances. Tous deux sont écrits dans une langue dont le vocabulaire peut être combiné d'une infinité de manières. Bien qu'un logiciel soit souvent une méthode de communication moins

directe que la prose, puisqu'il peut y avoir de nombreux intermédiaires entre un programmeur et l'utilisateur final d'une application qui utilise un morceau de son code, il en est de même pour d'autres formes d'expression. Au théâtre, par exemple, les spectateurs ne lisent pas directement le texte de la pièce, mais le voient et l'entendent au travers de la prestation d'intermédiaires (les acteurs) ; le texte n'en est pas moins un écrit.

Ni les essais, ni les logiciels ne sont écrits en portant une attention particulière à ce qu'a dit quelqu'un d'autre, et encore moins à ce qu'a décidé l'examineur officiel du bureau des brevets. Tous deux sont plutôt écrits dans la perspective de résoudre un problème particulier ou d'accomplir un objectif important pour l'écrivain. Dans les deux cas, des similitudes avec les travaux d'autres personnes se présentent naturellement à cause des similitudes entre les problèmes étudiés, et non à cause du plagiat des idées ou de la copie d'autres implémentations.

La programmation n'est pas plus ésotérique que le fût autrefois la prose, ou que l'est encore aujourd'hui la composition musicale. Il y a quelques centaines d'années, l'analphabétisme était courant. Acquérir la capacité d'écrire de la prose nécessitait un apprentissage. C'est toujours le cas, mais aujourd'hui, on enseigne l'écriture à tout le monde à l'école. D'autres formes d'écriture, comme la composition musicale ou le développement de logiciels, sont considérées optionnelles, mais nous les qualifions néanmoins d'écriture. Même si les notes ne chantent pas d'elles-mêmes &mdash; elles doivent être jouées &mdash; nous estimons que la composition musicale est une forme de discours ou d'expression.

De manière similaire, bien qu'un programme doive être exécuté pour être utilisé, avant de pouvoir être exécuté, il doit être écrit. Il y a actuellement des millions de gens aux États-Unis qui savent écrire un programme d'ordinateur. Il est absurde de s'attendre à ce que ces millions d'individus entament des recherches sur les brevets existants, ou n'importe quel autre type de recherche, avant de commencer à écrire un programme destiné à résoudre un problème spécifique. Si d'autres souhaitent acheter un logiciel, comme il en va de la vente de la prose écrite ou de la musique écrite, aucune restriction basée sur l'existence d'un brevet ne devrait entraver la capacité des auteurs à publier leurs propres écrits. Agir dans le sens d'une telle restriction procéderait de la confusion entre un piano (qui est brevetable) avec un arrangement de notes sur une partition spécifique (qui est effectivement soumise au droit d'auteur, mais non à loi sur les brevets).

De manière regrettable, les tribunaux ont laissé l'office des brevets promouvoir la censure de fait du travail de plus d'un million de développeurs professionnels, et de plusieurs millions d'amateurs. On demande maintenant à ces millions de citoyens de censurer leurs propres écrits, ou de les faire analyser et censurer par des tierces parties ou les tribunaux. Pour chaque brevet logiciel accordé, les auteurs de logiciels doivent donc quémander les ayants-droits de leur concéder &mdash; moyennant finance &mdash; les licences les autorisant à s'exprimer. Les détenteurs de ces brevets ne sont par ailleurs en aucun cas obligés d'accéder à ces demandes.

La suppression de la libre pensée et de la libre expression du logiciel (écriture, ou publication) est un mal, même si seul un petit nombre d'individus s'en rend compte : à combien se monteront les coûts si l'on peut recourir sans limite à cette censure insidieuse, déguisée sous le nom de « licence de brevet ».

En matière de logiciel, l'attribution par une agence gouvernementale d'un privilège exclusif de liberté d'expression, et la stigmatisation de toute expression alternative des mêmes idées (algorithmes) est intrinsèquement préjudiciable. En effet, cela revient à dire « N'essayez pas de résoudre des problèmes et d'inventer des solutions qui vous conviennent ; vous, ou vos programmeurs contractuels, pourriez, indépendamment, écrire ou inventer quelque chose que les examinateurs du bureau des brevets auraient placé sur la Liste des Algorithmes Interdits ; auquel cas, selon leur bon plaisir, ils pourront vous entraîner dans une procédure inquisitoriale coûteuse et traumatisante »...

D'après le Premier amendement, la liberté d'expression, par la parole ou l'écrit, ne peut être brimée par aucune branche du gouvernement ni par aucun représentant accrédité du gouvernement. Ceci prévaut quelles que soient les

intentions, bonnes ou mauvaises, de ceux qui présentent une argumentation contraire.

---

# La planification centralisée ou la soumission à un régime de licence des bonnes idées ne fonctionnera pas pour le logiciel

Exiger des programmeurs qu'ils aient connaissance de tous les brevets potentiellement applicables, revient au même qu'exiger des écrivains d'avoir lu toutes les oeuvres littéraires potentiellement significatives, avant même d'envisager de poser leur plume sur le papier. Ce n'est pas possible ; et quand bien même, ce ne serait pas désirable.

Le dommage principal occasionné par les brevets logiciels n'est pas passé ou présent ; il se situe dans l'avenir. Avec l'augmentation du nombre d'oeuvres logicielles et l'utilisation toujours plus large de la Programmation orientée-objet, des Algorithmes génétiques, des Réseaux de neurones et des techniques d'Ingénierie logicielle, le volume, la portée, la complexité et l'extension des logiciels écrits continueront leur développement exponentiel.

De manière concomitante, le pourcentage de tout ce qu'une seule et même personne pourra connaître par rapport à l'ensemble des connaissances informatiques diminuera exponentiellement. Ceci s'applique aussi bien aux programmeurs, qu'à ceux qui lisent les programmes, aux utilisateurs, et finalement aux examinateurs de brevets. En informatique, aujourd'hui déjà, personne ne peut connaître plus qu'une infime partie de ce qui est non-évident et innovant ; demain, le problème n'en sera que pire.

Nous entrons maintenant dans une ère où des programmes écrivent et modifient d'autres programmes, où il sera tout simplement impossible pour quiconque de savoir quels programmes ont évolués ou ont été automatiquement modifiés au point de ressembler étroitement à d'autres programmes, innovants ou non. De plus, nouveaux programmes et algorithmes seront écrits au rythme des logiciels s'exécutant pour les créer et non au rythme des programmeurs humains rédigeant le code ligne par ligne.

À ce stade, l'existence de brevets logiciels passera du statut d'improductif à celui d'impossible, comme le reconnaît avec raison une large majorité. Non seulement, les dispositions relatives aux brevets ne pourront être appliquées, il sera impossible de s'y conformer. Entre-temps, des sociétés réelles devront payer à des avocats réels des sommes toujours plus conséquentes pour tenter d'éviter des poursuites judiciaires, négocier des accords de licences croisées par ailleurs inutiles et perdre continuellement temps, argent et énergie dans ces activités défensives d'arrière-garde, qui n'apporteront strictement rien à la productivité des États-Unis ni au véritable fond des logiciels innovants.

Accorder des brevets logiciels est équivalent à demander à chaque programmeur de savoir ce que les millions d'autres programmeurs sur Terre sont en train de faire ou ont déjà accompli. Exiger une telle omniscience de la part des programmeurs est une manière évidente de les forcer à la désobéissance civile ; même s'ils n'en n'ont pas l'intention, il ne pourra en être autrement une fois qu'ils se rendront compte qu'ils violent inconsciemment un nombre croissant de brevets dont ils ne peuvent adéquatement être tenus au courant. Lorsque les lois commencent à dire aux gens de faire des choses impossibles, la désobéissance et le non respect de la loi sont une conséquence inévitable.

Ainsi, la seule chose que les brevets logiciels peuvent faire est de construire des barrières toujours plus arbitraires et coûteuses sur la route du progrès qui découle notamment de l'utilisation des nouveaux outils et techniques pour écrire des logiciels.

Étant donné que la promotion de l'innovation est la principale raison d'être des brevets, et puisqu'au contraire, en informatique, les brevets contrarient l'innovation et le feront de plus en plus, il est tout-à-fait clair que les brevets ne doivent pas être appliqués à l'écriture des logiciels. Cette seule raison est déjà suffisante, indépendamment des problèmes concernant le Premier amendement dont il a été question plus haut.

Des considérations similaires peuvent servir d'arguments contre une interprétation libérale du droit d'auteur, y compris dans les revendications étendues d'un « environnement graphique » (*look and feel*). La tâche de la preuve devrait toujours incomber au détenteur des droits, établissant clairement qu'il y a eu copie ou ingénierie inverse. Les programmeurs comme les écrivains ne peuvent en permanence être sur le qui-vive de peur qu'un jour quelqu'un puisse essayer de le poursuivre à cause de similitudes partielles ou accidentelles avec d'autres travaux. En l'occurrence, une interprétation libérale du droit d'auteur peut même être pire que le brevet, parce qu'une revendication d'infraction au droit d'auteur peut être soumise par presque n'importe qui et quel que soit le domaine, sans limitation temporelle significative, et sans aucune obligation de déclarer publiquement à l'avance quels aspects précis du travail sont suffisamment uniques pour prétendre à la protection du droit d'auteur. En conséquence, l'idée de la protection de droit d'auteur, cohérente quand il s'agit d'un écrit spécifique, devient incohérente et requiert de fait une omniscience lorsqu'on l'applique à toutes les variations possibles de travaux écrits.

Mais comme cette analyse se focalise sur la loi des brevets et le logiciel, la discussion des dangers de l'interprétation du droit d'auteur ne sera pas approfondie dans ce document.

---

## L'évolution des idées grâce à l'apprentissage par l'erreur est entravée par la brevetabilité des écrits

À chaque fois qu'un brevet sera attribué pour une expression particulière d'une idée en informatique, cela aura un effet dissuasif sur quiconque envisage d'écrire un logiciel pour résoudre des problèmes similaires. Il lui faut désormais procéder avec une extrême prudence, sous peine que ses écrits soient plus tard considérés comme couvrant partiellement le domaine dont la propriété exclusive a été attribuée au détenteur du brevet ; ceci, même si son expression a été déduite indépendamment et même si c'est une meilleure solution au même problème. Étant donné l'imprédictabilité manifeste des décisions de justice, nombreux sont ceux qui décideront prudemment de ne pas innover dans des domaines pour lesquels un brevet a déjà été attribué. Ceci est un moyen plutôt pervers d'encourager l'inventivité.

Le favoritisme d'un gouvernement envers des inventions logicielles « non évidentes » (c-à-d importantes) est comme le favoritisme du gouvernement envers des écrits religieux particuliers ou des produits commerciaux spécifiques : c'est intrinsèquement une mauvaise idée. L'idée selon laquelle les écrits de quelqu'un en particulier à propos d'un certain sujet seront certifiés par la branche exécutive du gouvernement, tandis que toutes les autres expressions seront supprimées, apparaîtrait très étrange aux fondateurs de la constitution.

Comme dans une oeuvre de fiction, la valeur d'un logiciel sophistiqué ne réside pas dans la simple idée de l'intrigue, mais dans la trame complexe du récit. Seuls ceux peu familiers des sentiments exacerbés, croyances et préférences qui existent chez les programmeurs à propos des différentes expressions des « mêmes » idées informatiques, peuvent croire que ces différences sont futiles pour ceux qui écrivent des logiciels, ou ceux qui utilisent des logiciels écrits par d'autres.

Imaginez que, pendant 17 ans, il ne soit permis qu'à un seul auteur d'écrire des histoires dont la ligne directrice serait « un garçon rencontre une fille, la fille quitte le garçon, le garçon regagne le coeur de la fille ». Ou qu'un jour, un quelconque consortium d'artistes ait inventé le rock and roll ou les quatuors à cordes (et ait produit une première «

réduction pour entraînement ») : personne d'autre ne pourrait écrire de musique dans ce style pendant 17 ans sans leur permission. Ou encore qu'une fois que le premier mathématicien aurait inventé une technique pour diviser les nombres, tous les autres mathématiciens devraient pendant 17 ans quémander la permission d'inventer leurs propres techniques, de peur de réinventer accidentellement ce qu'un autre a pu imaginer avant eux. Dans tous ces exemples, imaginez l'arrogance de la personne revendiquant le droit de traîner devant les tribunaux, pour les faire condamner, tous ceux qui choisissent de penser par eux-mêmes et d'écrire de manière indépendante.

Toute assertion défendant le droit d'un individu ou d'une organisation à établir l'exclusivité sur l'utilisation ou la spécialisation d'idées abstraites &mdash; évidentes ou non, importantes ou non &mdash; concrétisée dans une oeuvre de prose, musique, mathématique ou informatique, devrait troubler la conscience de quiconque dont le travail créatif est basé pour tout ou partie, comme c'est nécessairement le cas, sur des idées ou des techniques découvertes et développées par d'autres.

Perverse est la notion selon laquelle meilleure est l'idée, plus justifié est le choix d'accorder à son auteur un monopole sur toutes ses expressions possibles. Ce sont précisément les idées les plus importantes qui méritent l'exploration la plus complète et variée. Plus important ou innovant est le champ d'application, plus il est important d'éviter le favoritisme ou la censure de la part gouvernement de la pensée et de l'expression. Ne laisser le gouvernement attribuer un monopole que pour les idées nouvelles les plus importantes (« non évidentes ») ne rend pas le système meilleur mais pire.

En plus d'être préjudiciable à la liberté d'expression et d'expérimentation avec les idées exprimées par d'autres, les brevets logiciels sont également inutiles. Comme pour n'importe quel roman ou film un tant soit peu complexe, des logiciels conséquents atteignent un niveau de complexité et de détail significatifs. Hormis la copie directe, qui peut et doit être empêchée grâce à la législation sur le droit d'auteur, l'expertise et l'intelligence que nécessitent la création et l'amélioration continue d'un logiciel sophistiqué procurent une protection naturelle contre les concurrents potentiels ; ceux qui investissent dans l'écriture de tels logiciels maîtrisent les arcanes du problème qu'ils essaient de résoudre, bien mieux que ne le pourrait un pur copieur. Demander plus de protection, c'est cependant trop demander. Dans toutes les démarches qui jalonnent la vie, on fait des investissements que l'on aimerait pouvoir protéger. Mais quand la protection de quelques uns impose l'acquiescement d'un droit d'entrée aux autres, nous devrions toujours nous résoudre à l'abstinence. Un logiciel assez complexe et original pour mériter la protection d'un brevet n'en a pas besoin ; et un logiciel tellement simple qu'il a besoin de cette protection, ne la mérite pas. Comme dans le cas de la censure religieuse du dix-septième siècle, la question s'écarte ici des intentions initiales (les brevets logiciels en tant que stimulants de la créativité, etc.), mais plutôt, comment nous pouvons éliminer une erreur dangereuse mais évitable : un système de licence restrictive pour les logiciels ; et comment nous pouvons, au contraire, rétablir la totale liberté d'expression dans l'écriture et la publication des programmes informatiques.