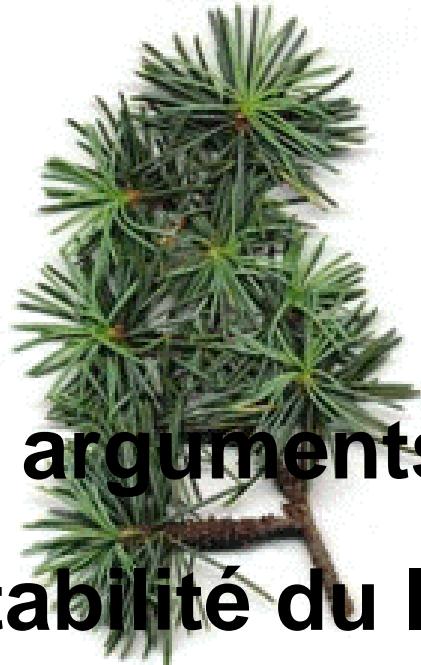


<https://laurentbloch.org/BlogLB/Quelques-arguments-contre-la>



Quelques arguments contre la brevetabilité du logiciel

- Zinformatiques - L'informatique : science et industrie - Logiciel libre -

Date de mise en ligne : dimanche 7 août 2005

Copyright © Blog de Laurent Bloch - Tous droits réservés

« Agir pour faire en sorte que tout ce qui est bien devienne bien commun et que tout soit libre pour ceux qui sont libres »
(**extrait de l'aphorisme 87 du Voyageur et son ombre**,
Friedrich Nietzsche).

Voici :

- Le brevet est destiné à protéger une idée de réalisation technique. La réalisation technique d'un dispositif matériel coûte beaucoup plus cher que le dessin qui le décrit, et qui permet de le breveter, c'est pourquoi il est raisonnable de protéger par un brevet les plans et les descriptions d'un tel dispositif, avant d'en entreprendre la construction. Ou la formule d'une molécule, avant de construire les installations qui permettront sa synthèse.
- Les idées contenues dans le logiciel ne sont pas des idées de réalisations techniques, mais les idées qui constituent le logiciel. Une fois que ces idées ont été écrites, elles sont le logiciel.
- Le brevet ne convient pas au logiciel, qui n'est qu'une suite d'idées : il faudrait un brevet toutes les trois lignes. Ce qui veut dire qu'avec des brevets sur les logiciels, lorsque l'on écrirait un logiciel, toutes les trois lignes on courrait le risque d'enfreindre un brevet.
- Lors des [Trophées du Logiciel Libre](#) 2005 à Soissons, j'ai entendu Jean-Paul Figer, de CAP-Gemini Ernst Young, déclarer : *le droit d'auteur ne coûte rien et protège un logiciel que l'on a réalisé, le brevet coûte fort cher et protège un logiciel que l'on n'a pas (encore) écrit.* À vrai dire, j'ai été agréablement surpris de me trouver si d'accord avec lui !
- Il y a des cas où l'on peut avoir une idée de la densité idéelle du logiciel. Ainsi, le noyau sécurisé du logiciel du métro METEOR a été développé en utilisant la méthode B, qui consiste à écrire, en même temps que le programme proprement-dit, la preuve de sa justesse, selon un formalisme spécial. La cohérence des deux est vérifiée automatiquement. Dans ce cas, la validation du logiciel en question, long de 100 000 lignes, a engendré 30 000 exigences de preuve. De ces 30 000 preuves, 27 500 furent démontrées automatiquement par le système, et 2 500 durent être démontrées « à la main » par des ingénieurs. Il est peut-être possible de dire que ces 100 000 lignes de programme comportent 30 000 idées, dont 2 500 idées difficiles.
- Pour les raisons énumérées ci-dessus, si les brevets sur le logiciel entrent en vigueur, développer du logiciel va devenir une activité très dangereuse, parce que toutes les trois lignes on risquera d'enfreindre un brevet détenu par un gros détenteur : IBM, Microsoft, Oracle et d'autres déposent des milliers de brevets par an, souvent rédigés de façon très générale et vague. La seule façon pour une PME d'échapper à des procès qui, même gagnés, la condamneraient à la faillite, ce serait d'avoir pour seuls clients IBM, Microsoft, Oracle, etc.

Pour reprendre une idée de Patrick Sinz, nous serions ainsi ramenés à la situation de l'industrie automobile, où il n'y a qu'une dizaine de grands constructeurs, entourés d'une myriade de sous-traitants et d'équipementiers. Ce qui serait une bonne façon d'annéantir le potentiel d'innovation et de foisonnement de l'industrie du logiciel, mais on voit bien l'intérêt qu'y trouveraient IBM, Microsoft, Oracle, etc.

Le 29 juin je faisais partie d'une délégation conviée à Bruxelles par la [Foundation for a Free Information Infrastructure](#) pour expliquer notre point de vue aux Euro-députés, qui étaient assez peu nombreux à nous écouter

Quelques arguments contre la brevetabilité du logiciel

(voir la [photo](#)), mais qui nous invitèrent à déjeuner(voir la [photo](#)). Nous étions cent-vingt, dont dix-sept Français. [Erwan Hamon](#), de [RotomalUG](#) (le groupe Linux de Rouen), a fait de cette expédition un [récit détaillé](#) dont je vous conseille la lecture palpitante. Sachant que le groupe de pression opposé était mené par Siemens, Philips et Nokia, nous étions assez pessimistes dans le Thalys de retour. Mais finalement le vote du 6 juillet fut une bonne surprise !

Pour d'autres idées sur cette question, on pourra lire un [article de Michel Volle](#), une [interview de Michel Rocard](#), une [contribution de Pierre Aigrain](#), et [un autre article du même](#), notamment. Voir aussi un texte général sur le [logiciel libre](#).

Voir aussi une analyse sarcastique : [Microsoft Files to Patent Emoticon Method](#).

Et ici [un document accablant](#) !

Post-scriptum :

Après le vote qui a conduit au rejet de la directive :

[L'analyse de la FFII](#), c'est-à-dire la [Foundation for a Free Information Infrastructure](#).

Le [discours de Michel Rocard](#).