

# Les applications industrielles et scientifiques des logiciels libres : aperçu général

## Mots clés

Logiciels libres,  
Gratuits,  
GNU, Licences,  
Copyleft,  
Brevetabilité des logiciels

■ Par Jean-Pierre HAUET  
*Associate Partner BEA Consulting- Président ISA-France*

## 1. Freeware et free software

La notion de « logiciel libre » prête souvent à confusion. Beaucoup s'imaginent qu'un logiciel libre est un logiciel gratuit, et il est exact que des logiciels libres le sont effectivement. Mais là ne réside pas leur caractéristique principale. Un logiciel libre n'est pas nécessairement gratuit. Par contre, un logiciel libre est par définition libre de droit d'usage et de modification.

L'ambiguïté provient du double sens qui s'attache au mot « free » en anglais :

- « free » peut signifier « gratuit » et des logiciels gratuits seront appelés « freeware » »,
- mais « free » peut aussi signifier « libre » et on débouche alors sur la notion de « logiciels libres » ou « free software » en anglais.

*“Freeware is a matter of liberty, not price – You should think of “free” as in “free speech”<sup>1</sup>.*

<sup>1</sup> Free Software Fondation : <http://www.fsf.org>

La notion de logiciel libre payant surprend. Pourtant, s'agissant surtout d'applications professionnelles, elle devrait rassurer. Un bien doté d'une valeur économique a normalement un prix, et les logiciels libres n'échappent pas à cette règle.

## 2. Les « freewares »

Les logiciels libres sont nés postérieurement aux « *freewares* » ou « *gratuits* » dont l'apparition remontent à 1982.

La notion de *freewares* a été introduite par Andrew Fluegelman, auteur d'un logiciel de communication, *PC-Talk*, qu'il ne souhaitait pas promouvoir et distribuer par des voies traditionnelles. Son modèle économique était en fait davantage celui du *shareware* (voir plus loin) que celui du *freewares* au sens où on l'entend aujourd'hui, c'est-à-dire :

- un logiciel dont l'auteur autorise gratuitement l'utilisation,

### L'ESSENTIEL

Apparu en 1984, le modèle économique des logiciels libres est souvent confondu avec celui des freewares, dont la gratuité est avant tout un acte commercial. Les logiciels « libres », au sens libres d'accès, procèdent d'une autre approche et ont vocation à valoriser les synergies entre concepteurs, développeurs et utilisateurs.

L'exemple-type est celui du système GNU-Linux qui a ouvert la voie à des dizaines de milliers d'autres logiciels dans les domaines les plus variés.

Les logiciels libres constituent un domaine de prédilection pour l'innovation, et sont aujourd'hui une alternative possible aux logiciels propriétaires, y compris dans les domaines les plus exigeants des applications industrielles et scientifiques. Toutefois, un certain nombre de précautions doivent être prises avant de les adopter. Par ailleurs, la question toujours pendante de la brevetabilité des logiciels reste une question fortement débattue, qui peut avoir une incidence sur leur avenir.

### SYNOPSIS

Since its apparition in 1984, the free software business model is often mixed up with the freeware one. Freeware responds to a commercial strategy while free software results from another approach and is aiming at capitalizing on synergies between designers, developers and users.

The typical example is GNU-Linux which led the way to dozens of thousands of other software products in the most various application areas.

The free software is a privileged field for innovation and is today a candidate option for replacing proprietary software, including in the most demanding area of industrial and scientific applications. However, a number of precautions must be taken before making a decision. Moreover, the issue of software patentability remains open and is widely debated. It may have a significant impact on the future of free software.

- tout en conservant les droits de propriété intellectuelle,
- sans généralement donner accès au code source,
- sans autoriser les modifications.

Les *freewares* sont fréquemment téléchargeables sur Internet. Il s'agit souvent de produits d'appel destinés à inciter à l'acquisition, payante cette fois, de logiciels interopérables avec eux. A titre d'exemple, on peut mentionner : *Adobe Reader*<sup>2</sup> et *Macromedia Flash Player*<sup>3</sup>.

Les *freewares* peuvent également servir de supports publicitaires aux fameux « *pop-up* » qui encombrant nos écrans. On parle alors de « *adwares* ». Mais ces *adwares* dégénèrent fréquemment en *spywares*, qui épient le comportement et les habitudes des utilisateurs à leur insu. Ces *spywares* peuvent eux-mêmes dégénérer en virus, véritables fléaux du monde informatique actuel.

Une variété importante de *freewares* est celle des *sharewares*. L'appellation *shareware* fut en fait utilisée afin de contourner les droits exclusifs d'exploitation qui avaient été acquis par *Andrew Fluegelman* sur celle de *freeware*, mais les *freewares* de *Fluegelman* étaient dans la réalité des *sharewares*, c'est-à-dire des ;

- logiciels mis gratuitement mais temporairement à la disposition des utilisateurs, à des fins de test ou de promotion commerciale,
- avec des droits d'usage limités dans le temps ou par le nombre d'utilisations,
- sans aucun transfert de propriété intellectuelle.

*WinZip*<sup>4</sup> ou l'outil de veille automatisée *KBcrawl*<sup>5</sup> sont des exemples de *sharewares*.

Parmi les *sharewares*, on pourra distinguer les *cripplewares* qui passent en *freewares* à l'expiration de leur période de test, mais avec des fonctionnalités réduites. L'éditeur de formules mathématiques, compatible *Word*, *Mathtype*<sup>6</sup> en est un exemple.

Comme on le voit, il y a toujours, derrière la notion de *freeware*, une approche marketing, plus ou moins sophistiquée, qui fait de la gratuité d'utilisation du logiciel, permanente ou définitive, un argument de promotion ou de vente pour le logiciel lui-même ou pour des produits associés.



Figure 1. Photo de Richard Stallman extraite de l'encyclopédie libre Wikipedia.

### 3. La notion de logiciel libre

Les logiciels libres ou *free softwares* s'inscrivent au contraire dans un certain idéal, souvent -mais pas toujours- désintéressé. Leur philosophie s'est construite vers 1984 autour du projet GNU, et peut se définir ainsi :

« *S'appuyer sur une communauté d'intérêts entre chercheurs, développeurs et utilisateurs afin de parvenir à des logiciels non assujettis aux contraintes des logiciels propriétaires* ».

Le projet *GNU* a été lancé en 1984 par *Richard Stallman*<sup>7</sup>, afin de développer un système d'exploitation complet, libre d'accès, compatible avec Unix.

*GNU* (prononcer guh-noo) est un acronyme récursif signifiant « *GNU's Not UNIX* ». Ce projet, très modulaire, connut une avancée décisive en 1991 avec l'apport de *Linus Torvalds*<sup>8</sup>, créateur du noyau *Linux*.

Aujourd'hui, l'ensemble *GNU-Linux* est souvent appelé « *Système GNU/Linux* ». Il est largement utilisé et fournit la base des "Distributions *Linux*".

La notion de logiciel libre a été théorisée par la FSF<sup>9</sup> (Free Software Foundation), créée en 1985 par *Richard Stallman*, qui a posé quatre principes de base (voir tableau 1).

<sup>2</sup> Adobe [www.adobe.org](http://www.adobe.org)

<sup>3</sup> Macromedia [www.macromedia.com](http://www.macromedia.com)

<sup>4</sup> WinZip Computing [www.winzip.com](http://www.winzip.com)

<sup>5</sup> BEA Conseil [www.beaconseil.com](http://www.beaconseil.com)

<sup>6</sup> Design Science Inc [www.dessci.com/en/company](http://www.dessci.com/en/company)

<sup>7</sup> Voir Sam Williams – “*Free as in freedom – Richard Stallman's crusade for free software*” – Editions O'Reilly - mars 2002.

<sup>8</sup> Voir Linus Torvalds –David Diamond – « *Il était une fois Linux – L'extraordinaire histoire d'une révolution accidentelle* » - Editions Osman Eyrolles Multimedia - mai 2001.

<sup>9</sup> [www.redhat.com](http://www.redhat.com) et [www.mandriva.com](http://www.mandriva.com)

1. Le droit d'exécuter sans contrainte le programme (liberté 0).
2. Le droit d'étudier comment fonctionne le programme et de l'adapter à ses besoins. Ceci implique l'accès au code source (liberté 1).
3. Le droit de redistribuer des copies afin d'aider d'autres utilisateurs (liberté 2).
4. Le droit d'améliorer le programme et de diffuser les perfectionnements afin d'en faire bénéficier tout utilisateur potentiel. L'accès au code-source est également un prérequis (liberté 3).

Tableau 1. Les principes de base des logiciels libres selon la FSF.

La notion de logiciel libre se réfère donc à la liberté, pour les utilisateurs, d'exécuter, de reproduire, de distribuer, d'étudier, de modifier et d'améliorer le logiciel.

Mais libre ne veut pas nécessairement dire gratuit. Le développement d'activités commerciales autour de logiciels libres : services de distribution, de maintenance, de formation etc. est parfaitement possible. Les sociétés *Red Hat* et *Mandriva (ex Mandrakesoft)*<sup>10</sup>, toutes deux distributrices de plates-formes Linux, en sont des illustrations.

Libre ne veut pas dire non plus exempt de licence : les utilisateurs néophytes de logiciels libres sont souvent surpris d'avoir à prendre connaissance et à respecter les termes d'une licence gouvernant l'utilisation d'un logiciel qu'ils pensaient a priori exempt de toute contrainte. Nous y reviendrons plus loin.

#### 4. Le copyleft

Dans la gestion des logiciels libres, la question des modifications est délicate : en l'absence de protection sur la propriété intellectuelle, un utilisateur peut s'appuyer sur une base réputée publique pour développer des améliorations, dont il fera un logiciel propriétaire. Pour pallier ce risque, la FSF a imaginé et développé le concept de « copyleft ».

Le « copyleft » permet de faire en sorte que les versions modifiées ou étendues d'un logiciel libre le soient également.

Le « copyleft » s'appuie sur le « copyright », mode usuel de protection des logiciels. Mais les conditions de distribution d'un logiciel libre « copylefté » imposent à chaque utilisateur apportant des modifications ou développant un logiciel dérivé, de distribuer, s'il décide de le faire, le produit de ces modifications ou améliorations dans les mêmes conditions que le logiciel initial.

Le « copyleft » correspond à une notion de transitivité. Mais il souligne l'importance des licences dans le domaine des logiciels libres aussi bien que dans celui des logiciels propriétaires.

#### 5. Les dérives idéologiques et la création de l'Open Source Initiative (OSI)

On l'aura compris : la notion de « logiciel libre » ne correspond pas à une définition légale. La FSF a cherché à « s'approprier » de facto le contrôle de l'appellation au travers de la licence GNU GPL (General Public Licence)<sup>11</sup>. Mais le mouvement GNU, héritier des idées libertaires des années 1970, est apparu à certains trop idéologique, trop politique, voire trop sectaire.

Ainsi est née en 1998, l'Open Source Initiative<sup>11</sup> visant à donner au mouvement du « Libre » un caractère plus industriel et commercial. L'Open Source promeut des concepts qui sont très voisins de la FSF, et logiciels libres et logiciels en Open Source sont souvent confondus. Il est vrai que les logiciels libres au sens FSF sont en règle générale Open Source, mais l'inverse n'est pas toujours vrai, au regard notamment du critère relatif à la distribution des perfectionnements.

Pour pouvoir se réclamer de l'Open Source et utiliser son logo (Figure 2), un logiciel doit satisfaire à 10 critères :



Figure 2. Logo de l'Open Source Initiative.

1. Droit de libre redistribution, sans imposition de royalties
2. Accès au code source libre et facile
3. Droit de modification et possibilité de distribuer les œuvres dérivées dans les mêmes termes que le logiciel initial
4. Respect du droit moral de l'auteur de l'œuvre initiale (traçage et nommage des œuvres dérivées)
5. Pas de discriminations vis-à-vis des personnes et des groupes
6. Pas de restrictions sur les champs d'applications
7. Licence applicable à tous, sans restrictions additionnelles
8. Licence non spécifique à un produit particulier
9. La licence ne doit pas générer de restrictions sur d'autres logiciels
10. Licence technologiquement neutre

Tableau 2. Les 10 critères définissant un logiciel en Open Source selon l'OSI.

<sup>10</sup> Voir [www.gnu.org/licenses](http://www.gnu.org/licenses)

<sup>11</sup> [www.opensource.org](http://www.opensource.org)

## 6. Les licences

Les tenants de la FSF et du « Libre », au sens natif du terme, sont souvent prompts à dénoncer des abus d'utilisation du qualificatif de « Libre », dont personne ne peut cependant revendiquer l'exclusivité.

Nous pensons qu'il faut en la matière, s'agissant d'applications industrielles, adopter une approche pragmatique et non dogmatique, fondée essentiellement sur les objectifs que l'on s'est donnés.

Dans cet esprit, les dispositions de la licence accompagnant la distribution d'un logiciel libre doivent toujours être examinées avec soin, surtout si l'on souhaite le modifier ou l'intégrer dans un produit matériel ou logiciel destiné à être commercialisé.

La FSF a homologué au regard de ses critères, outre la licence de base GPL déjà citée, environ 25 licences jugées compatibles avec la GPL et 35 licences non compatibles. L'OSI a approuvé environ 60 licences. Les licences les plus connues sont :

- GPL
- Lesser GPL
- Berkeley (BSD)
- MIT
- Mozilla (MPL)

Le CNRS, l'INRIA et le CEA proposent quant à eux la licence CeCILL, intégrant certains points particuliers du droit français.

## 7. Pourquoi utiliser des logiciels libres, où les trouver, comment les choisir ?

- L'intérêt économique et financier est souvent la préoccupation la plus immédiate. Mais, même en l'absence de droits d'utilisation, il faudra, avant de prendre une décision, faire un bilan d'ensemble.
- Les mises à jour sont souvent plus fréquentes.
- La robustesse des logiciels peut être meilleure du fait de retours d'expériences d'un grand nombre d'utilisateurs. Cependant, à cet argument, les professionnels du logiciel opposent souvent l'amateurisme de beaucoup de développeurs des communautés du « Libre ».
- Une meilleure sécurité contre les « *backdoors* » et les risques d'intrusion voire de prise de contrôle à distance.
- Une ouverture plus large sur l'innovation, sans volonté sous-jacente d'imposer une remise à niveau payante de la base matérielle et logicielle préexistante.

*Toutefois le modèle économique du "Libre" reste fragile. La pérennité des produits peut être une préoccupation, ainsi que leur maintenance. La question des services associés est essentielle. L'intérêt économique du "Libre" peut évidemment s'en trouver amoindri.*

Il faut donc porter une grande attention au choix des logiciels avant de les retenir pour des applications industrielles ou scientifiques. La palette est extrêmement large et 70 000 logiciels libres étaient, paraît-il, accessibles à la mi-2004.<sup>12</sup>

On trouve des logiciels libres répondant à toutes sortes de préoccupations :

- des systèmes d'exploitation : *Linux* restant le plus connu
- des outils de programmation : *Perl*
- des outils de bureautique : *OpenOffice* et la suite *Internet Mozilla* : *Firefox, Thunderbird, SPIP, Apache* etc.
- des logiciels d'éducation, de sciences
- des utilitaires (antivirus)
- des jeux, des logiciels multimédias etc...

On peut dire aujourd'hui que le domaine du « Libre » n'a plus de limites.

Ces logiciels sont accessibles à partir de sources diversifiées :

- la presse spécialisée publie souvent des dossiers comparatifs ou thématiques,
- les annuaires tenus par certaines organisations proposent des listes, souvent triées par chapitre. La FSF en propose, en collaboration avec l'UNESCO, 3 975 ; Framasoft 881<sup>13</sup> ; l'OSI propose une sélection dans les domaines des OS, de l'Internet et des outils de programmation,
- Les forums et les sites de rencontre sur Internet : *Freshmeat*<sup>14</sup>, *SourceForge*<sup>15</sup>, *Open Source Development Network*<sup>16</sup> peuvent être consultés.

Enfin, dans le monde du « Libre », le « bouche à oreille », les expériences, les tuyaux que l'on se communique de façon désintéressée, jouent un rôle essentiel.

L'abondance même de l'offre constitue un problème en soi. Elle implique une analyse sérieuse avant de faire un choix, surtout pour des applications industrielles ou scientifiques de haut niveau.

Un processus de sélection rigoureux est nécessaire. Il faut bien sûr tester le logiciel et s'assurer de ses fonctionnalités et de sa convivialité. Mais il faut aussi examiner la

<sup>12</sup> Perline et Thierry Noisette – *La bataille du logiciel libre* – Editions la découverte – Oct 2004.

<sup>13</sup> [www.framasoft.net](http://www.framasoft.net)

<sup>14</sup> <http://freshmeat.net>

<sup>15</sup> <http://sourceforge.net>

<sup>16</sup> <http://osdn.net>



crédibilité et les références du logiciel candidat, Internet étant la source d'informations privilégiée.

Il faut également s'assurer de l'existence de services autour du produit et notamment vérifier qu'il existe, autour du logiciel, une « communauté » de développeurs crédible.

*La communauté de développeurs est l'éditeur "virtuel" qui assure, en respectant les principes du « Libre », la pérennité et l'évolution d'un logiciel donné. Elle doit comprendre des développeurs, des testeurs, des rédacteurs, des traducteurs, etc. Du sérieux de cette organisation dépend le degré de confiance que l'on peut accorder à un logiciel libre.*

Aujourd'hui, les administrations, les collectivités territoriales sont des utilisateurs privilégiés en raison des avantages économiques qui s'attachent aux logiciels libres.

Les éditeurs de logiciels incorporent souvent dans leurs produits des briques logicielles libres.

Enfin les mondes industriels et scientifiques viennent progressivement aux logiciels libres, après les avoir souvent longuement testés, étant séduits par leur diversité, leur degré d'innovation et leur indépendance par rapport aux grands éditeurs.

Les constructeurs d'équipements informatiques ne dédaignent plus ces solutions, et leur consacrent dans leur offre une place qui va au-delà du témoignage, et qui contribue à leur crédibilisation face à des solutions solidement implantées.

## 8. Le problème des brevets logiciels

Nous terminerons ce tour d'horizon en évoquant le problème des brevets logiciels, un sujet qui passionne et divise le monde du « Libre ». La communauté Linux/FSF lui est farouchement hostile, l'OSI lui est plutôt favorable.

On sait que les logiciels ne sont pas brevetables en tant que tels, en France et dans beaucoup de pays. Le droit d'auteur leur est applicable et il est recommandé de les déposer de façon officielle chez un notaire ou à l'Agence pour la Protection des Programmes (APP)<sup>17</sup>. Mais le droit d'auteur ne protège que contre la copie servile : le code reste la propriété de l'auteur, alors que l'algorithme dont il est la traduction peut être utilisé librement.

Depuis plusieurs années, l'Office Européen des Brevets (OEB) délivre des brevets logiciels si le logiciel est la traduction d'une solution nouvelle et non évidente d'un problème technique, et si cette solution est définie par des caractéristiques techniques.

Les USA ont quant à eux une interprétation beaucoup plus large de la brevetabilité des logiciels, et les grands éditeurs de logiciels déposent chaque année des milliers de brevets.

Dans ce contexte, une Directive communautaire sur la brevetabilité des inventions mises en oeuvre sur ordinateur a été mise en discussion en 2002. L'idée était, de façon résumée, d'officialiser la doctrine de l'Office européen des brevets.

Le débat très large qui s'en est suivi a été passionné, et a donné lieu à des débats très vifs entre le Parlement européen, le Conseil et la Commission.

Le 7 mars 2005, après un premier vote hostile du Parlement, le Conseil adoptait un texte qui devait revenir dans les trois mois devant le Parlement en seconde lecture.

Finalement, le **6 juillet 2005**, le Parlement européen a rejeté la position commune du Conseil des ministres avec 648 votes pour le rejet et seulement 14 contre. La Commission européenne avait annoncé auparavant qu'en cas de rejet par le Parlement, elle ne représenterait pas le texte. Plusieurs points restent en suspens cependant : parmi l'écrasante majorité qui a voté contre la directive, certains eurodéputés l'ont évidemment fait parce qu'ils préféraient le *statu quo* à une prohibition explicite des brevets logiciels, et l'Office européen des brevets peut continuer à accorder des brevets à des logiciels, malgré la Convention de Munich.<sup>18</sup>

Le débat, du seul fait de la mondialisation du problème, ne nous semble pas clos. Nous ne pensons pas que les idées sous-jacentes au projet de Directive aient pu être préjudiciables aux logiciels libres, aux petites et moyennes entreprises, ou à l'effort d'innovation en général. Il nous semble qu'à une époque où une fraction importante de la capacité inventive s'exprime par la voie de logiciels, il est souhaitable que soient reconnus les apports en propriété intellectuelle que constituent les logiciels. Une telle reconnaissance ne pourrait en outre qu'éviter les situations ambiguës relatives à la propriété de certains logiciels libres. Mais le débat reste ouvert.

### L' a u t e u r

**Jean-Pierre Hauet**, Ingénieur en Chef des Mines, a dirigé le centre de recherches de Marcoussis d'Alcatel avant d'être directeur Produits et Techniques de Cégélec. Il a été nommé Chief Technology Officer d'ALSTOM, lors de l'acquisition de Cégélec par cette dernière. Depuis 2003, il est consultant, Associate Partner de BEA Consulting. Il préside l'ISA-France, section française de l'ISA.

<sup>17</sup> Agence pour la Protection des Programmes - <http://app.legalis.net>

<sup>18</sup> Ce § est extrait de Wikipedia, encyclopédie libre : [http://fr.wikipedia.org/wiki/Brevet\\_logiciel](http://fr.wikipedia.org/wiki/Brevet_logiciel)