

Java, .NET et les logiciels libres

Fabrice Rossi
<http://apiacoa.org/>

Publication originale : décembre 2004
Version actuelle : 21 juillet 2005

Cet article a été publié dans le numéro 67 de la revue GNU/Linux Magazine France en Décembre 2004. Je remercie Denis Bodor et les éditions Diamond pour avoir autorisé la distribution de cet article selon les conditions indiquées dans la section A. L'article est disponible en ligne à l'URL <http://apiacoa.org/publications/2004/lm-java-dotnet-free.pdf>. La section B détaille les modifications effectuées dans l'article depuis sa publication. J'ai pour objectif de faire évoluer l'article dans les mois qui viennent afin de le maintenir à jour. Si vous avez des suggestions ou des commentaires, merci d'utiliser le journal correspondant sur mon compte linuxfr¹.

Résumé

Cet article présente les problèmes liés au contrôle, par l'intermédiaire de brevets logiciels et d'accord de licence complexes, de Java et de .NET par les entreprises qui les ont créés. Il tente d'apporter des éléments de réponse à une question fondamentale pour la communauté du libre : Java et .NET sont-ils compatibles avec le logiciel libre ?

¹Url : <https://linuxfr.org/~boubou/>

1 Introduction

Deux plate-formes de développement remportent un succès important dans les entreprises : Java et .NET. Elles comportent de nombreux points communs, en particulier celui d'être justement des cadres de développement (des plate-formes) qui allient un langage orienté objet de haut niveau (Java ou C#), une plate-forme d'exécution (les machines virtuelles) et une bibliothèque gigantesque qui couvre à peu près tout les besoins d'un programmeur. Mon but dans cet article n'est pas de m'intéresser aux aspects techniques de Java et de .NET, mais plutôt de me focaliser sur leurs ressemblances les plus gênantes pour les auteurs de logiciels libres (ou open source) : Java et .NET ont été créés par des entreprises privées (par opposition aux langages issus de la recherche académique, par exemple) et leurs implémentations de référence ne sont pas des logiciels libres ou open source. De plus, divers brevets logiciels sont de la partie. Je me propose d'étudier ici les conséquences de ces points cruciaux, en oubliant presque totalement le mérite technique des deux plate-formes.

Avant de me plonger dans les problèmes de licences et de brevets logiciels, j'aimerais cependant faire un petit détour par la technique. En effet, si Java et .NET n'avaient aucun intérêt technique, il n'y aurait pas lieu de

se poser la difficile question de leur compatibilité avec le libre. Nous disposons de nombreux langages et bibliothèques à la fois libres et d'excellente qualité, et nous n'avons donc pas vraiment besoin de langages supplémentaires, même si la diversité n'a jamais été un problème pour le libre, au contraire. Même dans cette hypothèse, il nous faudrait cependant prendre garde de ne pas nous couper du reste du monde, car Java et .NET sont de plus en plus utilisés dans les entreprises. L'enseignement de la programmation, dans les écoles spécialisées et les universités, ne peut s'affranchir de ce fait. En pratique, Java est donc utilisé comme langage d'apprentissage de la programmation dans de très nombreuses formations, et il est probable que C# le rejoigne dans les années à venir.

Au delà de l'aspect pragmatique, Java et .NET méritent notre intérêt car ils ne manquent pas d'atouts techniques. Il me semble que le plus important d'entre eux est la productivité qu'on peut avoir avec ces plate-formes. Celle-ci est avérée, établie par de nombreuses études, et c'est en général l'argument principal qui décide les entreprises à utiliser Java ou .NET. Cette productivité trouve ses racines dans la gestion automatique de la mémoire (le garbage collecting), dans le modèle orienté objet simple (sans héritage multiple), dans la qualité des environnements de développement (comme Eclipse, <http://www.eclipse.org/>, pour n'en citer qu'un), dans la complétude des API, etc. Pour développer des logiciels libres, la portabilité binaire (un des arguments de Java et .NET pour séduire les entreprises) n'est pas très utile, mais tout le reste, et en particulier la productivité accrue, est particulièrement séduisant pour notre communauté. A tel point que certains leaders du projet Gnome, comme Havoc Pennington par exemple, posent ouvertement la question (cf [1] par exemple) : l'avenir de Gnome passe-t-il par l'abandon pro-

gressif du C/C++ au profit de Java et/ou de C# ? Il ne s'agirait pas bien sûr de remplacer l'infrastructure de Gnome par une version Java/C#, mais plutôt de privilégier ces langages pour les nouvelles applications. Comme je l'ai déjà écrit, mon but ici n'est pas de répondre à cette question du point de vue technique, mais d'étudier les conséquences que pourrait avoir la généralisation de l'utilisation de .NET et de Java dans les projets libres, en déterminant qui contrôle ces plate-formes et ce que cela implique pour le libre. Je commencerai l'article par un bref rappel sur les éléments techniques nécessaires à la compréhension du reste du texte, puis j'étudierai le cas de Java avant de passer à .NET.

2 Brefs rappels sur les deux plate-formes

Dans cet article, je désigne par Java la plate-forme [2] qui porte ce nom, et qui est constituée du langage Java, de la machine virtuelle associée (la Java Virtual Machine, JVM) et de l'ensemble des API. Un programme écrit en Java est en général compilé vers le byte code de la machine virtuelle. Lors de l'exécution du programme, la JVM peut soit interpréter les instructions du byte code en les traduisant en des instructions du processeur cible, soit compiler au vol des morceaux de byte code en code objet natif (on appelle cette compilation au vol du Just In Time) puis les exécuter. Les implémentations les plus efficaces de la JVM mélangent en général l'interprétation et le JIT. Le passage par un byte code indépendant de la machine cible de l'exécution permet d'obtenir une forme de portabilité binaire.

Dans cet article, je désigne par .NET ce que Microsoft appelle le framework .NET. Cette plate-forme [3] ressemble beaucoup à Java. La principale différence est qu'elle se veut

neutre par rapport au langage de programmation et est donc construite à partir d'une machine virtuelle associée à un ensemble d'API. L'ensemble forme la Common Language Infrastructure (CLI). .NET lui-même consiste en la CLI auquel on ajoute divers langages (C#, Visual Basic .NET, etc.) et des bibliothèques complémentaires. La machine virtuelle de .NET (le Common Language Runtime, CLR) est très proche de la JVM, bien que les opérations choisies soit plus adaptées au JIT qu'à l'interprétation. De ce fait, les implémentations du CLR sont en général exclusivement basées sur du JIT. En pratique, l'utilisation du langage C# [4] est presque indispensable si on veut profiter des intérêts de .NET, bien que d'autres langages profitent aussi pleinement de la plate-forme, au prix de quelques adaptations (on peut citer Visual Basic et Eiffel par exemple). C# est très proche de Java. En caricaturant, on pourrait dire que c'est une version améliorée de Java, bien que les aficionados de ce dernier mettront en avant certains points discutables de C# (le modèle de la redéfinition de méthode est assez complexe, les exceptions ne doivent pas obligatoirement être traitées, etc.) pour refuser cette définition. Pour éviter toute polémique stérile, disons que C# et Java sont presque identiques, au moins vus depuis les autres langages. Les lecteurs intéressés trouveront dans le présent numéro de leur magazine préféré un article très complet d'introduction à C# proposé par Grégoire Lejeune.

Les deux plate-formes ne sont pas "pures", au sens où leurs implémentations contiennent en général du code qui n'est pas écrit dans le langage principal qui leur est associé. En fait, elles comportent toutes deux des mécanismes d'interfaçage qui permettent d'appeler des bibliothèques externes écrites dans divers langages. Ceci permet de réaliser une implémentation d'une plate-forme en s'appuyant sur l'existant, en particulier les bibliothèques

système. Il y a un certain consensus autour du fait que les mécanismes d'intégration de l'existant à .NET sont à la fois plus simples d'utilisation et plus complets que leurs équivalents en Java.

3 Le cas de Java

3.1 Java : pas de standard !

La plate-forme Java a été créée par la société Sun Microsystems qui a rendu disponible la version 1.0 de son implémentation en 1996. Java est une marque déposée par Sun (et on devrait donc écrire Java™ à chaque utilisation...), ce qui signifie qu'on ne peut pas utiliser librement le terme Java pour désigner un programme ou encore un langage (cf [5] pour des détails). A titre de comparaison, Linux est aussi une marque déposée et son utilisation n'est donc pas libre. Il faut obtenir une autorisation du Linux Mark Institute [6] pour certaines utilisations.

Historiquement, Sun a été tenté par la standardisation pour la plate-forme Java. Un dossier avait d'abord été déposé auprès de l'ISO (International Organization for Standardization, [7]) qui avait approuvé le principe de la création d'une norme Java en 1997 (cf [8]). Les discussions avaient cependant été dès le départ assez tendues entre l'ISO et Sun, car Sun ne voulait en aucun cas abandonner la marque Java et le contrôle de la plate-forme (cf [9]). En raison de ces tensions, Sun décida de passer par l'ECMA [10], ce qui lui aurait permis de déposer auprès de l'ISO le standard obtenu, sans passer par les comités techniques de l'ISO, grâce une procédure fast track (cf [9]). Cependant l'histoire s'est répétée. Bien que Sun ait déposé une proposition de standard à l'ECMA en mai 1999 (cf [11, 12, 13]), il est rapidement devenu clair que le passage de l'ISO à l'ECMA ne changeait pas grand chose. Comme l'ISO, l'ECMA n'est pas une

simple chambre d'enregistrement et voulait donc imposer son point de vue à Sun, en réalisant éventuellement des modifications de la proposition de standard Java. En fait, Microsoft, membre de l'ECMA, souhaitait proposer dans ce cadre des modifications à la spécification de Sun (cf [14]). Finalement, Sun décida fin 1999 d'abandonner ses efforts de standardisation (cf [15, 16]). Il n'existe donc pas de standard international pour Java.

3.2 Le Java Community Process

Parallèlement à cette tentative de standardisation, Sun a créé dès 1998 le Java Community Process (JCP, [17]). Il s'agit d'un programme de coopération autour de Java dont le but est d'assurer la maintenance de la plate-forme de façon collaborative. Toute l'évolution de Java est gérée par des votes des membres du JCP et s'articule autour des Java Specification Requests (JSR). Un JSR est une proposition d'extension de l'API de Java, de modification du langage ou de la JVM, plus généralement d'évolution de la plate-forme. La participation au JCP est payante pour les entreprises et les associations, gratuite pour les individus. Malgré le caractère ouvert du JCP, les décisions fondamentales sont prises par l'Executive Committee (EC) seul. L'EC est composé d'un membre de droit (Sun), de 5 membres élus et de 10 membres nommés. Les membres élus le sont au sens habituel du terme, alors que les membres nommés sont désignés par le bureau du JCP (le Program Management Office) qui est lui même formé par Sun. Les nominations sont approuvées par un vote. Le fonctionnement du JCP est décrit par [18], document qui est lui même modifiable par le JCP!

Sun a donc cédé une partie de son contrôle sur Java par l'intermédiaire du JCP. Les membres qui possèdent vraiment le pouvoir (l'EC) sont cependant essentiellement des

grandes entreprises (comme IBM, SAP ou même SCO) avec deux exceptions notables à savoir la fondation Apache et le professeur Doug Lea de l'Université de New York Oswego (cf [19]). De plus, Sun conserve un droit de veto sur l'inclusion de modifications dans les éditions "standard" de la plate-forme Java (il existe trois éditions, la version Standard, la version Entreprise destinée aux serveurs et la version Micro destinée à l'embarqué).

Enfin, rejoindre le JCP signifie signer le JSPA (Java Specification Participation Agreement), un contrat entre Sun et le nouveau membre : Sun peut donc parfaitement refuser de signer le contrat et donc ne pas accepter d'accueillir un candidat dans le JCP. Le contrat est assez complexe (cf [20]) et détaille les obligations et droits des membres du JCP, en particulier en terme de brevets et de "propriété intellectuelle". Deux points importants sont à retenir :

1. quand un JSR est approuvé, sa spécification est obligatoirement publiée sur le site du JCP, mais elle reste la propriété du leader du JSR ;
2. quand un JSR est approuvé, le leader et ses collaborateurs doivent obligatoirement accorder une licence sans royalties (Royalty Free) pour l'implémentation du JSR par quiconque voudrait réaliser une telle implémentation. Cette licence doit couvrir tous les brevets concernés et doit être perpétuelle et irrévocable. Par contre, elle ne s'applique qu'à des implémentations conformes, point que nous étudierons en détail dans la section suivante.

De prime abord, la situation semble donc assez claire. Si je souhaite implémenter Java 1.5, la dernière version, il me suffit de me rendre sur le site du JCP et de chercher la spécification correspondante. Il s'agit en fait du JSR n°176 [21] dont la spécification est tout

simplement la documentation du kit de développement de Sun [22]. Une lecture (passionnante;-) de la licence de la spécification [23] confirme ce que j'ai indiqué précédemment : on peut implémenter librement Java 1.5, à condition que l'implémentation soit conforme. On constate sur le site de Sun qu'on peut implémenter la version 1.4.2 de Java dans les mêmes conditions, mais que les anciennes versions ont des licences beaucoup plus restrictives.

3.3 Est-on vraiment libre ?

En fait, l'impression de liberté qu'on peut ressentir en lisant les lignes qui précèdent doit être tempérée en raison de divers problèmes.

Historiquement tout d'abord, le JCP était beaucoup plus fermé. L'ancienne version du JSPA ne permettait pas vraiment des implémentations open source. Devant les critiques, en particulier en provenance de la fondation Apache (cf [24, 25, 26]), Sun a réformé le JSPA pour obtenir la version actuelle que j'ai décrite au dessus (cf [27]). De plus, Sun s'était engagé à l'époque à appliquer les nouveaux termes (en particulier la possibilité d'implémentations libres) aux JSRs qu'il avait dirigés, ce qui a été fait. Cependant, Sun n'est pas le leader dans tous les JSR, loin s'en faut. Il est parfaitement possible que les leaders de certains JSR débutés avant l'entrée en vigueur du nouveau JSPA (fin 2002) continuent à appliquer les processus de l'ancien JSPA. Cela ne représente pas un gros problème pour une implémentation libre de la plate-forme Java puisque la partie fondamentale est gérée par Sun (et que les versions 1.4.2 et 1.5 sont implémentable librement). Cependant, cela signifie que certaines API officielles de la plate-forme pourraient ne pas être implémentées librement.

L'aspect le plus gênant pour le libre ne réside cependant pas dans ces "détails", mais

plutôt dans la définition d'une implémentation conforme. En effet pour être légale, une implémentation

1. doit implémenter complètement la spécification ;
2. ne doit pas ajouter, modifier ou supprimer quelque chose dans les packages définis par la spécification ;
3. doit passer avec succès les tests de compatibilité.

Concernant le point 3, il faut savoir qu'un JSR est obligatoirement associé à un Technology Compatibility Kit (TCK), c'est-à-dire un ensemble de tests qui couvrent normalement toute la spécification produite par le JSR. Une implémentation qui ne passe pas le TCK n'est donc pas légale. Or, le TCK est loin d'être accessible librement. Il est en général payant (et très cher), on ne peut pas le redistribuer, le modifier, etc. (cf [26] pour la licence accordée à la fondation Apache). Fort heureusement, la version actuelle du JSPA encourage les leaders de JSR à offrir gracieusement leur TCK à toute organisation à but non commercial et à tout particulier. Sun donne l'exemple en offrant ainsi le TCK de certains JSR à la fondation Apache (pour l'implémentation des servlets et des JSP par Tomcat, par exemple, cf [26]) et à ObjectWeb (pour JOnAS, cf [28]).

Contrairement à ce qu'on pourrait croire, le point 3 n'est finalement pas le plus gênant pour le libre. En fait, ce sont les points 1 et 2 qui posent problème : si je télécharge la spécification du JDK de Sun, je n'ai plus le droit de proposer une implémentation partielle de celui-ci. Légalement, je ne peux que proposer une implémentation complète (on imagine le travail de titan que cela demande, vue la complexité et la taille de l'API Java et sachant que les anciennes spécifications de Java, plus simples que les versions actuelles ont des licences beaucoup plus restrictives) sans avoir réalisé de modification dans les

API. Donc, adieu CVS et autre développement ouvert ! Richard Stallman a bien expliqué le dilemme que cette situation crée dans [29]. Le plus sournois dans cette histoire est que une fois une implémentation libre développée, rien n'empêche de distribuer des versions modifiées qui ne respectent pas la spécification, cela est même explicitement indiqué dans l'accord de licence. Celle-ci dit en effet qu'on peut distribuer une implémentation conforme de Java sous une licence qui n'inclut pas les trois clauses de conformité. On peut alors imaginer le scénario suivant :

1. un développeur X télécharge la spécification d'une API Java (exemple la fondation Apache se procure la spécification des servlets) et doit donc accepter la licence correspondante (licence Sun) ;
2. le développeur X implémente scrupuleusement (de façon fermée) l'API et diffuse une implémentation conforme (Tomcat dans notre exemple) sous une licence libre classique (licence A) ;
3. un développeur Y (différent de X), qui n'a pas téléchargé la spécification de l'API, se procure l'implémentation de celle-ci (toujours Tomcat) et accepte donc la licence A ;
4. le développeur Y modifie l'implémentation de sorte qu'elle ne soit plus conforme et la diffuse, toujours en accord avec la licence A (sous le nom de Jerrymouse, par exemple).

A l'issu de ce scénario, on se retrouve avec un logiciel libre (Jerrymouse) qui implémente quelque chose qui ressemble à l'API Java de départ, sans pour autant être conforme, et tout ça légalement. La sacro-sainte compatibilité qui justifie pour Sun les restrictions autour de Java peut donc parfaitement être contournée, modulo le problème des brevets logiciels.

3.4 Java et les brevets logiciels

On pourrait se demander dans quelle mesure l'implémentation de Java à partir de livres et d'autres sources autres que Sun ne permettrait pas de contourner les restrictions imposées par Sun plus directement que dans le scénario qui précède. On peut même lire la documentation du jdk directement sur le site de Sun, sans accepter de licence. Il y a malheureusement un piège dans cette approche : les brevets logiciels ! En effet, Sun possède des brevets qui couvrent certains aspects de Java, bien que leur liste n'ait jamais été rendue publique (cf [24], page 5). Or, pour obtenir une licence Royalty Free pour ces brevets, il faut accepter la licence de la spécification concernée. De ce fait, redistribuer une implémentation libre d'une spécification de Sun n'est peut être pas légal car cela peut éventuellement correspondre au non respect d'un brevet de Sun, même si on n'a pas téléchargé la spécification correspondante (cf le point de vue de Havoc Pennington à ce sujet dans [30]). Sauf bien sûr si l'implémentation est conforme ! Nous voici donc revenus à la case départ.

Le plus amusant avec les brevets logiciels, c'est qu'ils peuvent se retourner contre vous. Sun vient en effet de verser 92 millions de dollars US à Eastman Kodak Co. pour régler un problème de brevets logiciels s'appliquant à Java (cf [31] pour les trois brevets concernés). L'histoire complète est décrite sur divers sites (cf [32, 33, 34, 35, 36]). En résumé, Kodak a attaqué Sun en février 2002 en l'accusant d'utiliser des algorithmes et principes brevetés dans Java. Les principes en question sont plutôt vagues et semblent s'appliquer à bien d'autres choses que Java puisqu'ils tournent autour de l'idée de collaboration entre objets par le biais de bus logiciel (il est donc probable que les brevets en question s'appliquent à .NET). Le 1er octobre 2004, un jury américain s'est prononcé en faveur de Kodak. La

cour ne se prononçait pas sur la validité des brevets de Kodak (il pourrait très bien exister des mise en oeuvres antérieures des idées brevetées), mais sur la violation de ceux-ci par Sun dans la plate-forme Java. Sun reconnu coupable, Kodak s'est empressée d'indiquer qu'elle allait demander 1,06 milliards de dollars US en dédommagement soit plusieurs années de profits de Sun. Moins d'une semaine plus tard (le 7 octobre), Sun annonçait qu'un accord était trouvé, pour un montant de 92 millions de dollars US. Bien entendu, les détails de l'accord ne sont pas connus, excepté le fait que Sun a obtenu une licence pour l'utilisation des brevets incriminés dans Java [37]. La formulation du communiqué de presse de Sun laisse entendre que ses clients sont protégés par l'accord. Des officiels de Sun ont indiqué clairement que les produits basés sur Java dans le cadre d'un accord de licence avec Sun sont aussi couverts par l'accord avec Kodak. Une implémentation officielle de Java est donc couverte, même si elle est réalisée sous licence open source par quelqu'un d'autre que Sun. Cependant, l'attitude de Kodak sur le long terme n'est pas facile à prévoir, et le moins qu'on puisse dire est que l'accord n'a pas totalement rassuré les adeptes de Java qui auraient largement préféré que Sun aille en appel ou que les brevets soient invalidés (cf [38, 39] pour des interprétations de l'accord Sun Kodak).

Pour clore cette discussion sur les protections de Java mise en place par Sun, on peut évoquer le droit des marques. En effet, toute implémentation de Java passe par l'écriture de classes et interfaces qui appartiennent aux packages `java` et `javax`. Dans quelle mesure la programmation de `java.lang.String` (les chaînes de caractères), par exemple, correspond-elle à une utilisation de la marque Java ?

3.5 Les implémentations libres de Java

La situation de Java est donc loin d'être claire et il n'est pas évident que la réalisation d'une implémentation libre de Java au sens où nous l'entendons habituellement soit vraiment légale, tout au moins pendant son développement. Pourtant de nombreux projets libres existent autour de Java. Je vous épargnerai une longue liste pour me focaliser sur quelques cas importants. La plate-forme Java comportant un langage, une machine virtuelle et un ensemble de bibliothèques, les initiatives libres couvrent en général une partie seulement de ces éléments :

- Jikes [40] est un compilateur libre pour le langage Java, produit par IBM. Il transforme des sources Java en byte code. Jikes est complet au sens où il est capable de compiler toutes sources Java jusqu'à la version 1.4, la mise à niveau vers 1.5 étant en cours.
- Classpath [41] est un projet GNU officiel qui vise à implémenter l'ensemble des bibliothèques de la plate-forme Java dans son édition standard. Classpath n'est pas une implémentation complète (et n'est donc pas légale!), mais elle réalise des progrès constants.
- Il existe de nombreuses machines virtuelles open source. Pour fournir une plate-forme complète, elles utilisent en général Classpath ou une version modifiée de celle-ci. Les JVM suivantes sont activement développées (liste absolument non exhaustive) :
 - Jikes RVM [42] est une JVM écrite par IBM en Java (si, si) avec presque aucun code C (elle utilise Classpath) ;
 - Kaffe [43] est une des plus anciennes implémentations libres de Java ; elle utilise en partie Classpath, en partie des bibliothèques spécifiques ;

- SableVM [44] est une JVM issue de la recherche académique (comme Jikes RVM), qui contient de nombreuses innovations (comme une implémentation très intéressante des verrous de synchronisation qui évite d'utiliser un mot par objet), et qui s'appuie sur Classpath ;
- IKVM.NET [45] est une JVM assez fascinante car elle est implémentée en .NET ! Elle permet donc de faire fonctionner des programmes Java à partir d'un environnement .NET comme celui de Microsoft ou une implémentation libre de celui-ci. IKVM.NET utilise Classpath.
- Le projet GCJ [46] propose de compiler du Java directement vers du code natif, afin d'améliorer les performances en évitant l'interprétation ou le JIT. GCJ est intégré à GCC et utilise une version modifiée de Classpath.
- Notons enfin que certaines API sont implémentées par des projets indépendants de Classpath. C'est le cas en particulier pour tout ce qui touche à l'aspect entreprise pour lequel la fondation Apache, le consortium ObjectWeb et le groupe JBoss sont très actifs. En fait, contrairement à la partie couverte par Classpath, les API entreprises sont totalement implémentées sous forme de logiciel libre (JBoss et JOnAS en particulier). JBoss a ainsi passé officiellement le TCK pour la partie entreprise de Java 1.4 (cf [47]). En théorie, JBoss est donc couvert par l'accord entre Sun et Kodak sur les brevets logiciels de ce dernier, car JBoss a bien un accord officiel de licence avec Sun pour l'implémentation de Java 1.4 version entreprise.

Malgré le travail réalisé sur diverses implémentations libre de Java, il est clair qu'il n'existe pas pour l'instant d'alternative com-

plète à l'implémentation propriétaire de Sun (dont certains éléments sont eux-même open source, comme par exemple la partie chargée de l'analyse de documents XML ou encore le moteur de servlet/JSP, à savoir Tomcat). Cependant, de grands progrès ont été réalisés en 2004, à tel point qu'on peut maintenant faire tourner avec GCJ des programmes très complexes comme l'environnement de développement Eclipse.

On peut noter que pour l'instant, Sun n'a pas réagi de façon négative aux différentes initiatives open source alors même que le développement de celles-ci se fait de façon complètement ouverte. Des tensions ont existé avec le groupe JBoss, mais Sun a plutôt fait preuve de bonne volonté (en particulier en aidant la fondation Apache et ObjectWeb), sans pour autant accepter de rendre son implémentation de Java libre, malgré des pressions incessantes de nombreux acteurs du monde Java, IBM en tête.

4 Le cas de .NET

4.1 Les normes ECMA/ISO

La situation de .NET beaucoup plus simple que celle de Java. En fait, .NET est en partie standardisé, en partie totalement contrôlé par Microsoft son principal créateur (la version 1.0 de .NET est sortie en janvier 2002).

En août 2000, deux mois après le lancement officiel de C# (en juin 2000), Microsoft a déposé les spécifications du langage C# et de la CLI auprès de l'ECMA pour obtenir une standardisation (cf [48]). On peut noter au passage que C# et la CLI sont une oeuvre commune de Microsoft, Intel et HP. En décembre 2001, l'ECMA a ratifié une première version des spécifications, puis une deuxième version un an après (cf [49] et [50]). Les standards ECMA ont été soumis à l'ISO qui les a ratifiés en avril 2003 (cf [51]). Depuis, l'ECMA tra-

vaille à la mise à jour des standards, en particulier la spécification de C# 2.0 (cf [52]) qui intègre des nouveautés importantes comme les types paramétrés (les generics).

La norme ECMA-334 couvre l'intégralité du langage C# (dans sa version 1.0) alors que la norme ECMA-335 s'intéresse à la CLI, c'est-à-dire la machine virtuelle (le CLR) et la bibliothèque standard. Un point crucial à noter est que cette bibliothèque est assez limitée et que la CLI n'est que la brique de base de .NET. En fait, la CLI contient 7 bibliothèques fondamentales qui couvrent les besoins classiques, à savoir les entrées/sorties (fichier et réseau), les collections, la manipulation de documents XML, la gestion des chaînes de caractères, tout le mécanisme de réflexion (l'introspection), etc. On est très loin d'avoir ici l'intégralité de .NET, comme nous le verrons dans la section suivante.

Contrairement à de trop nombreuses normes (les standards POSIX par exemple), les normes C# et CLI sont librement et gratuitement accessibles sur le site de l'ECMA (cf [49] et [50]). Les conditions de redistribution de ces documents ne sont malheureusement pas indiquées.

4.2 Les bibliothèques sans standard

.NET ne serait pas à même de concurrencer Java ou d'autres plate-formes de développement s'il se limitait aux bibliothèques incluses dans la CLI. En fait, .NET comporte trois grandes bibliothèques cruciales dans la plupart des applications :

1. Windows.Forms : il s'agit de la bibliothèque qui permet de créer les interfaces utilisateurs. Grossièrement, c'est donc l'équivalent .NET de Gtk, de Qt ou encore de Swing et SWT pour Java.
2. ADO.NET : il s'agit de la bibliothèque d'accès aux données stockées dans une base de données, l'équivalent .NET de ODBC ou de JDBC.
3. ASP.NET : il s'agit des bibliothèques " serveurs " de .NET. Elles permettent entre autres de créer des sites web dynamiques et des services web (dans le même esprit que PHP et les JSP).

Microsoft contrôle totalement ces bibliothèques et plus généralement tout les aspects de .NET qui ne sont pas standardisés par l'ECMA. Il s'agit essentiellement de bibliothèques additionnelles (cf [53] pour une vue générale des bibliothèques de .NET). Pour mesurer l'étendue de ce contrôle, on peut noter que la CLI contient 350 types (classes, interfaces, structures, etc.), contre 3800 types dans .NET fin 2002 (cf [54]).

4.3 Peut-on implémenter librement une norme ?

Beaucoup de gens pensent naïvement qu'une norme officielle peut être implémentée librement. Et pourtant, rien n'est moins vrai. En fait, chaque organisme de normalisation décide des restrictions qu'il juge acceptable pour les normes qu'il produit. Un exemple célèbre est celui du format de compression sonore MP3. MP3 est en fait la couche audio de niveau 3 des standards vidéo MPEG-1 et 2. MPEG est un groupe de travail issu de l'ISO et de l'IEC (International Electrotechnical Commission), dont la vocation est de créer des standards pour la compression vidéo. Or, pour créer MP3, le groupe a utilisé des travaux de l'un de ses membres, le Fraunhofer Institute. Associé à Thomson Multimedia, le Fraunhofer Institute possède 18 brevets touchant de près ou de loin au format MP3 (cf [55, 56]). Thomson exige actuellement le paiement de royalties pour toute utilisation commerciale de MP3 (cf [57]). Il faut bien entendu signer un accord de licence officiel pour

pouvoir utiliser légalement (et commercialement) MP3, au moins dans les pays qui reconnaissent les brevets logiciels. Ce problème de brevets autour de MP3 a été historiquement une des motivations à la création du format Ogg et du schéma de compression associé Vorbis (cf [58]).

La politique de l'ECMA vis à vis des brevets est très claire (cf [59]) : l'ECMA ne standardise des technologies couvertes par des brevets que si ceux-ci peuvent être licenciés en des termes "raisonnables et non discriminatoires" (RAND pour Reasonable And Non Discriminatory). En pratique, cela signifie que la licence ne doit pas être trop "chère" (à l'échelle d'une entreprise, bien sûr!) et que ces termes doivent être les mêmes pour tout le monde. L'ISO appliquant le même raisonnement, on peut supposer que les conditions commerciales proposées par Thomson sont RAND (2 500 \$ US pour utiliser MP3 dans un jeu, par exemple).

4.4 .NET et les brevets logiciels

Le problème est bien entendu que Microsoft possède des brevets relatifs à .NET. L'entreprise a même déposé en 2002 une demande de brevet sur l'intégralité de l'API de .NET de l'époque (cf [60, 61]). Pour savoir s'il est possible d'implémenter librement les normes C# et CLI, il faut donc connaître la position de Microsoft en terme de licences pour les brevets concernés. Or, il semble qu'il n'existe pas vraiment de position officielle de Microsoft à ce sujet (par exemple publiée sur le site web de Microsoft), mais seulement des indices que nous allons analyser dans la suite de l'article.

On connaît en fait deux indices concernant directement .NET. Premier indice, Jim Miller, un des inventeurs des brevets sur .NET, a indiqué en février 2003 dans une liste de diffusion consacrée à .NET (cf [62]) que Microsoft et ses co-inventeurs (Intel et HP) étaient prêts

à offrir des licences Royalty Free ou RAND pour les brevets concernant C# et la CLI (ceci est confirmé sur le site de Microsoft [63] par un engagement général concernant la participation à la création de standards).

Deuxième indice, Michele Herman, la responsable de tout ce qui concerne la "propriété intellectuelle" chez Microsoft, a indiqué dans un entretien accordé à David Berlind de ZDnet (cf [54]), que Microsoft accorderait une licence RAND aux personnes qui souhaiteraient implémenter .NET. Cette licence sera aussi Royalty Free. Cela signifie qu'il faudra un accord formel avec Microsoft, mais que cet accord n'impliquera pas de contribution financière. Herman a bien indiqué que l'accord en question interdirait les sous-licences : en d'autres termes, pour obtenir une licence, il faudra toujours s'adresser à Microsoft directement. D'après Herman elle-même [63], ceci est en totale contradiction avec la GPL (cf aussi l'analyse très intéressante de Seth Nickell [64]), et ceci de façon délibérée.

En effet, quand on parle de licences pour les brevets de Microsoft, il ne faut jamais oublier que l'entreprise est totalement opposée à la GPL. Bill Gates a notamment déclaré en Juin 2001 que la GPL était comparable au célèbre personnage de jeux vidéo Pac Man et qu'elle dévorait tout autour d'elle (cf [65, 66]). Le site de Microsoft contient d'ailleurs une présentation assez "amusante" de la GPL (cf [67]). Il semblerait que l'entreprise n'ait pas compris le principe de la double licence utilisée entre autres par Trolltech et MySQL AB, puisque la page citée indique qu'un logiciel diffusé sous GPL ne peut pas être commercialisé. Quelles qu'en soient les raisons, l'attitude de Microsoft est claire depuis plusieurs années : l'entreprise fera tout pour s'opposer à la GPL, tout en acceptant dans une certaine mesure les autres licences open source.

La politique de défense par Microsoft de sa "propriété intellectuelle" est cependant

complexe. Pour essayer de la comprendre, je vais présenter plusieurs exemples.

4.4.1 L'affaire du Sender ID

Le Sender ID (cf [68]) est une technique anti-spam développée entre autre par Microsoft dont les détails techniques importent peu ici. Le point crucial est que Microsoft a fait une demande de brevet concernant Sender ID et exige la signature d'une licence (de type Royalty Free) pour toute implémentation de celui-ci (cf [69] pour la licence). La licence proposée est considérée comme incompatible avec une implémentation libre par la fondation Apache [70] et Debian [71], entres autres. L'IETF [72], qui était censé produire un standard basé sur le Sender ID, a finalement abandonné ses travaux, rejoignant le consensus qui s'est formé dans le monde libre contre la licence de Microsoft. La seule exception à ce consensus est Sendmail qui a décidé d'implémenter Sender ID en raison d'une interprétation divergente de la licence de Microsoft et surtout avec un argument assez intéressant (cf [73]) : Microsoft a fait une demande de brevet concernant Sender ID, mais le brevet n'a pas encore été accordé (la situation est donc très semblable à celle des api de .NET). Notons que les aventures du Sender ID ne sont pas terminées, puisque devant les réactions de la communauté open source, Microsoft a décidé de revoir sa copie dans certaines proportions, en particulier en limitant la portée de sa demande de brevet (cf [74] pour une analyse de ce revirement).

Il ne faudrait cependant pas placer trop d'espoir dans la nouvelle attitude de Microsoft. En effet, comme l'indique la position officielle de la fondation Apache [70], il y a eu de nombreuses discussions, pendant des mois, entre des acteurs du libre (le juriste Larry Rosen en tête) et Microsoft au sujet de la licence proposée, avec une intervention de Michele

Hermann du côté de Microsoft, sans qu'un accord puisse finalement être atteint.

Le point à retenir dans l'exemple du Sender Id est donc que certaines licences Royalty Free de Microsoft sont incompatibles avec une implémentation open source (et pas seulement GPL), ce qui laisse craindre le pire pour le RAND.

4.4.2 Samba

Samba est une implémentation libre sous licence GPL du protocole SMB/CIFS qui permet le partage de fichiers et les services d'impression dans le monde Windows. CIFS est la dernière incarnation d'un protocole issu de NetBIOS, initialement développé par IBM et Sytec, puis amélioré par Microsoft, IBM et d'autres. Nous sommes donc en présence d'une technologie au moins en partie inventée par Microsoft et qui peut donc être considérée comme sa " propriété ". Le parallèle avec .NET est évident.

Pourtant, la situation pratique n'est pas totalement identique. Samba a en effet été développé en grande partie par reverse engineering, ce qui est parfaitement légal (au moins dans certains pays, dont l'Australie, terre patrie de Samba) quand on vise l'interopérabilité (cf à ce sujet [75]). Il est clair en effet que pour faire fonctionner une machine Unix dans un environnement Windows, il est presque indispensable de la faire " parler " SMB/CIFS. L'argument d'interopérabilité fonctionne donc parfaitement et justifie le reverse engineering, qui lui même évite de signer un quelconque accord de licence pour accéder à la documentation de SMB/CIFS.

Cependant, certains autres éléments rapprochent CIFS de .NET. Il existe en effet une sorte de spécification de CIFS, librement accessible (cf [76]). Cette documentation mentionne explicitement qu'elle ne s'occupe pas de l'aspect " propriété intellectuelle ". Son

existence ne signifie donc pas qu'on puisse implémenter CIFS librement (comme pour les normes relative à .NET en somme). D'autre part, certaines parties de CIFS sont couvertes par des brevets possédés par Microsoft. Cependant, d'après la Samba Team, les brevets en question sont spécifiques à l'implémentation de CIFS sous Windows et couvrent des parties obsolètes du protocole (cf [77]), ce qui signifie que Samba n'utilise pas les méthodes brevetées (cf aussi [78]).

Comme pour de nombreuses technologies Microsoft, il est possible d'obtenir une licence Royalty Free pour les brevets qui concernent CIFS. Bien que la licence autorise une implémentation open source, sa formulation (cf [79]) est incompatible avec la GPL, de façon totalement délibérée (on voit ici les déclarations de Michele Herman mises en pratique). On peut noter que la première version de cette licence interdisait même plus explicitement une implémentation sous GPL (cf [80]), en qualifiant cette dernière de " Intellectual Property Rights Impairing Licence ", c'est-à-dire de licence qui ne préserve pas les droits de " propriété intellectuelle " !

Il serait naïf de croire qu'on peut conclure de l'exemple de Samba que tout se passera bien pour les implémentations libres de .NET. En effet, Samba a été essentiellement construit par reverse engineering alors que les implémentations libres de .NET sont construites à partir des normes ECMA et des documentations Microsoft (pour les bibliothèques qui ne sont pas standardisées). De plus, SMB/CIFS est protégé par deux brevets américains et un brevet européen (cf [80]) dont les concepts ne sont pas utilisés par Samba, alors que Microsoft a demandé une protection de l'API de .NET par brevet, ce qui signifie que toute implémentation de .NET sera nécessairement couverte par le brevet en question (si celui-ci est accepté). Enfin, bien que Microsoft ait longtemps construit

son succès sans s'appuyer de façon décisive sur les brevets logiciels (cf [81]), cette époque est révolue, de même d'ailleurs que la condescendance à peine polie de l'entreprise envers les logiciels libres. Samba profite en quelque sorte du passé, car les premiers développements de SMB sont suffisamment vieux pour être contemporain de l'époque " sans brevet " de Microsoft. Ce n'est malheureusement pas le cas des technologies plus récentes.

N'oublions pas, par exemple, que l'auteur de VirtualDub (un logiciel d'édition vidéo, sous licence GPL et tournant sous Windows [82]) a été obligé de supprimer en 2000 le support du format ASF de son logiciel suite à une demande de Microsoft (cf [83]). Pourtant, l'auteur de VirtualDub avait compris le format ASF uniquement par *reverse engineering*, mais ce format est protégé par un brevet... D'ailleurs, le format est maintenant documenté sur le site de Microsoft, mais il est toujours interdit d'en faire une implémentation open source (cf [84] et l'accord contenu dans la spécification). Le support du format asf dans xine, MPlayer, videoLAN, etc. est possible car les parties problématiques sont développées dans des pays qui ne reconnaissent pas (pour l'instant) les brevets logiciels.

4.4.3 Une tentative de synthèse

Les exemples analysés précédemment montrent que Microsoft ne semble pas totalement opposé à des implémentations open source de certaines de ses technologies et est donc disposé à offrir des licences pour les brevets correspondants. Cependant, Microsoft est opposé à la GPL et formule ses licences de sorte à exclure toute implémentation sous cette licence. De plus, la politique de Microsoft est très variable selon les technologies concernées et peut aller jusqu'à l'interdiction pure et simple d'une implémentation open

source, d'une façon directe comme pour ASF ou indirecte comme pour Sender ID.

L'étude de la partie du site de Microsoft consacrée à la " propriété intellectuelle " (cf [63]) en particulier la zone consacrée aux standards [69] montre qu'il n'existe pas pour l'instant² de licence officielle pour C# et la CLI, ce qui est relativement inquiétant et apporte de l'eau au moulin des personnes qui craignent que .NET soit un cheval de Troie contre linux (cf [64]).

Il reste cependant difficile de savoir quelle sera sur le long terme la politique de Microsoft dans ses rapports avec le logiciel libre. En 2002, Microsoft planifiait une attaque en règle contre les logiciels libres construite sur son portefeuille de brevets (cf [85]). Pour l'instant, les attaques n'ont pas eu lieu, mais il est clair que Microsoft souhaite valoriser ses brevets (cf [86]), ce qui inquiète certaines figures emblématiques de l'open source (cf [87]). Microsoft demande par exemple le paiement de royalties pour l'utilisation du système de fichiers FAT dans certaines circonstances [88]. Le brevet principal est en passe d'être invalidé (cf [89]), mais si cela n'avait pas été le cas, Microsoft aurait très bien pu étendre ses demandes de royalties à toutes les utilisations de la FAT, en particulier donc à son implémentation sous Linux. Autre exemple édifiant, l'exclusion d'OpenOffice.org de l'accord de réconciliation signé entre Microsoft et Sun en avril 2004 (cf [90]) : Microsoft a le droit de poursuivre le projet pour toute infraction à l'un de ses brevets (alors qu'il n'a pas ce droit pour StarOffice!).

Notons que, contrairement au cas de Java, il n'y a pas eu pour l'instant de jugement autour de .NET et on ne sait donc pas vraiment si des brevets sont applicables à cette plateforme. La demande de brevet sur les API de .NET n'a pas encore été acceptée et il est diffi-

cile de prédire l'issue qu'elle va connaître. Par contre, l'affaire Kodak vs Java ne sera pas nécessairement sans conséquence pour les implémentations libres de .NET. En effet, les brevets concernés sont si larges qu'ils couvrent très certainement .NET. Or, d'après [38, 39], Microsoft a déjà négocié un accord avec Kodak pour utiliser les technologies couvertes par les fameux brevets. Si Microsoft n'a pas réussi à la faire, on ne voit pas comment une implémentation libre de .NET pourrait les contourner.

Pour finir sur une note plus optimiste, il faut garder à l'esprit que les brevets logiciels sont souvent accordés en dépit du bon sens et sans tenir compte de l'état de l'art au moment de la demande. Il est donc souvent possible d'exhiber des exemples de mise en oeuvre antérieure de ce qui a été inventé par les auteurs d'un brevet (on parle alors de prior art), ce qui invalide le brevet ou provoque son transfert à l'inventeur réel de ce qui est protégé (dans le cas de la loi des USA). De plus, un brevet ne protège pas les effets d'une invention (on peut protéger l'aspirine, pas le concept de lutte contre les maux de tête). Il est donc parfois possible de mettre en oeuvre un concept sans enfreindre les brevets qui sont sensés protéger ce concept (un point illustré par Samba, par exemple), au moins dans certains cas.

4.5 Les implémentations libres de .NET

La situation de .NET est donc assez peu claire, et même inquiétante si on est suspicieux de nature. Pourtant, des implémentations libres de .NET existent, même si leur légalité n'est pas certaine. On compte en fait deux implémentations très avancées :

- Mono [91] est certainement l'implémentation libre de .NET la plus avancée. Elle est sponsorisée par Novell et di-

²en juin 2005.

rigée par Miguel de Icaza. Mono 1.0 (lancé en juin 2004) est assez complet et comporte en particulier un compilateur C# et une implémentation de la CLI. A ceci s'ajoute une implémentation de certaines bibliothèques additionnelles de Microsoft comme ADO.NET et ASP.NET, un binding de Gtk pour .NET (Gtk#) et d'autres bibliothèques spécifiques Mono. Par contre, les Windows.Forms ne sont pas implémentées de façon complètes (loin s'en faut), ce qui limite singulièrement les possibilités d'exécution de code .NET développé pour avec la version de Microsoft. Miguel de Icaza, le leader du projet, est une personnalité controversée, connu pour ses déclarations souvent provocatrices. A l'occasion de la sortie de Mono 1.0, de Icaza a donné à Cnet un entretien assez intéressant (cf [92]). On retrouve dans cet entretien les provocations habituelles de de Icaza (il dit par exemple qu'avant la sortie de Mono, il n'y avait pas d'environnement de développement intégré moderne sous Unix...), mais aussi des éléments de réponse au sujet des interrogations quant à la légalité de Mono. Il précise que Novell a conduit une étude des brevets actuels de Microsoft qui a conclu que Mono n'enfreignait aucun d'entre eux (les brevets de Kodak ne sont pas évoqués). Il indique que si dans les années à venir on découvrirait que du code de Mono utilise sans autorisation un algorithme breveté, ce code serait soit ré-écrit pour contourner le problème, soit supprimé (cf aussi le blog de de Icaza, [93, 94]). On retrouve ce point de vue sur le site de Mono (cf [95]) et plus généralement chez les développeurs de Mono (cf [96, 97]). Nous n'avons plus qu'à espérer pour Mono que la demande de brevet déposée par Microsoft portant sur l'API entière de .NET ne

soit pas acceptée, car on voit mal comment on pourrait la contourner...

- Portable.NET est une implémentation partielle de .NET réalisée par le projet DotGnu [98]. Cette implémentation est moins complète que celle de Mono, en particulier en terme de bibliothèques additionnelles (par rapport à la CLI implémentée complètement). Par contre, il semble que pour l'instant le support de Windows.Forms soit plus complet dans Portable.NET que dans Mono.

Comme dans le cas de Java, les implémentations libres de .NET sont en retard par rapport à son implémentation propriétaire, en particulier sur le support des bibliothèques graphiques. On retrouve ici un parallèle parfait entre Windows.Forms pour .NET et Swing pour Java qui ne sont supportés que partiellement par les implémentations libres. Dans les deux cas, des bibliothèques concurrentes et libres permettent de réaliser des applications graphiques (Gtk# dans le cas de .NET et SWT, le toolkit graphique du projet Eclipse, dans le cas de Java), mais au détriment de l'aspect standard (Gtk# et SWT sont portables donc cet aspect n'est pas mis en cause par leur utilisation).

5 Conclusion

Étant donné les faits présentés dans cet article, il est bien difficile de répondre à la question " est-il légal de proposer une implémentation libre de Java ou de .NET ? " et à la question connexe " quel risque prend-on en développant avec Java ou .NET ? " N'étant pas juriste, je me garderais bien de proposer une interprétation des accords de licence, des lois sur les brevets, etc. Je ne peux que proposer un point de vue personnel d'amateur (éclairé par les faits présentés dans cet article).

Il me semble qu'il est possible (mais diffi-

cile) de faire une implémentation libre légale de Java dans ses éditions standard 1.4.2 et 1.5 et dans son édition entreprise 1.4. Pour obtenir une licence d'implémentation, il suffit de se connecter sur le site de Sun, de télécharger la spécification concernée et d'accepter l'accord proposé. Cet accord indique explicitement que la distribution d'une implémentation conforme ne demande pas une licence obligeant à conserver la conformité de l'implémentation, ce qui permet d'utiliser une licence libre. En pratique, la restriction de conformité est vraiment très dure à respecter et relativement incompatible avec un développement ouvert. Je ne sais donc pas si Classpath est une initiative qui respecte les lois. C'est en fait peu probable car l'accord de licence spécifie que l'implémentation doit passer le TCK. Or, pour obtenir le TCK, il faut nécessairement signer un accord de licence (très restrictif celui-ci) avec Sun, ce qui n'a pas été fait (pour l'instant au moins) pour Classpath. Rien ne permet donc de prétendre que Classpath est une implémentation conforme et elle ne respecte donc pas la licence Sun. Par contre, JBoss, Tomcat, JOnAS, etc. sont des implémentations parfaitement légales de certaines parties de Java, puisque les groupes qui ont produit ces implémentations ont signé un accord avec Sun pour obtenir les TCKs correspondants (cf [26, 28, 47] par exemple). Sun est bien entendu informé du caractère ouvert du développement de ces implémentations et sait donc que JBoss et JOnAS ont été diffusés dans des versions non conformes, ce qui n'a pas empêché les signatures d'accords. Classpath pourrait donc parfaitement devenir légal dans l'avenir. Notons cependant, même si cela est vrai pour presque tout logiciel de nos jours, que des brevets divers peuvent s'appliquer à Java sans que Sun n'en soit propriétaire, ce qui limite la portée pratique de l'accord de licence offert sur le site de Sun. Les détails de l'accord avec Kodak restent par

exemple secrets et il est donc relativement difficile de se prononcer sur cet aspect du problème.

Concernant .NET, je ne peux pas avoir une conclusion aussi "positive". Il est impossible aujourd'hui (début novembre 2004) d'obtenir des informations officielles de Microsoft concernant les termes de la licence RAND qui doit nécessairement couvrir C# et la CLI. Les licences RAND (et même Royalty Free) peuvent parfaitement être incompatibles avec les licences libres (ou au moins avec certaines licences libres), comme l'a montré le fiasco du Sender ID ou comme le montre la licence actuelle de SMB/CIFS. De ce fait, je ne sais pas si Mono et Portable.NET sont des implémentations légales de .NET. De plus, la demande de brevet déposée par Microsoft et portant sur l'intégralité de l'API de .NET pourrait rendre totalement incontournable la nécessité d'obtenir d'une licence, au moins dans les pays qui reconnaissent les brevets logiciels. Comme Seth Nickell [64], je pense donc que Mono, bien que fascinant du point de vue technique, présente potentiellement un danger pour le logiciel libre : en s'appuyant fortement sur cette plate-forme, on prendrait le risque de voir s'effondrer son implémentation libre et par conséquent toutes les applications développées avec elle. Mais surtout, dans un monde qui reconnaît les brevets logiciels, la légalité de Mono est dans les mains de Microsoft, une entreprise qui n'est pas connue pour ses sentiments bienveillants à l'égard du logiciel libre.

Une fois de plus, comme le disait Nicolas Boulay dans le numéro 50 du GNU/Linux Magazine, on constate que les brevets logiciels sont "une épée de Damoclès suspendue au dessus des logiciels libres". Il convient donc plus que jamais de participer à la lutte contre leur adoption en Europe [99].

Références

- [1] Havoc Pennington. Java, Mono, or C++ ? <http://ometer.com/desktop-language.html>, March 2004. Une réflexion sur le futur de GNOME.
- [2] Java sur la Wikipedia. [http://fr.wikipedia.org/wiki/Java_\(langage\)](http://fr.wikipedia.org/wiki/Java_(langage)) et http://en.wikipedia.org/wiki/Java_programming_language.
- [3] .NET sur la Wikipedia. http://en.wikipedia.org/wiki/Microsoft_.NET. La version française est vraiment très succincte : <http://fr.wikipedia.org/wiki/.NET>.
- [4] C# sur la Wikipedia. http://en.wikipedia.org/wiki/C_Sharp_programming_language.
- [5] Sun trademark and logo usage requirements. <http://www.sun.com/policies/trademarks/>, February 2005. La politique de Sun pour ses marques déposées.
- [6] Site web du Linux Mark Institute. <http://www.linuxmark.org/>.
- [7] Site web de l'International Organization for Standardization (iso). <http://www.iso.ch/>.
- [8] Tim Clark. Sun wins Java ISO approval. <http://news.com.com/2100-1001-205430.html>, November 1997. CNET News.com.
- [9] Stephen Shankland. Sun revising plan for Java recognition. <http://news.com.com/2100-1001-225197.html>, April 1999. CNET News.com.
- [10] Site web de ECMA International. <http://www.ecma-international.org/>.
- [11] Stephen Shankland. Sun renews Java standards effort. <http://news.com.com/2100-1001-225464.html>, May 1999. CNET News.com.
- [12] Stephen Shankland. Sun seeks control of Java process. <http://news.com.com/2100-1001-225500.html>, May 1999. CNET News.com.
- [13] Sun submits Java technology for industry standardization. <http://www.sun.com/smi/Press/sunflash/1999-05/sunflash.990506.1.html>, May 1999. Annonce de presse de Sun.
- [14] Stephen Shankland. Java standard vote opens door for Microsoft. <http://news.com.com/2100-1001-227769.html>, June 1999. CNET News.com.
- [15] Stephen Shankland. Sun reverses plan for Java standard. <http://news.com.com/2100-1001-234061.html>, December 1999. CNET News.com.
- [16] Stephen Shankland. Sun misses Java standard deadline. <http://news.com.com/2100-1001-233940.html>, September 2000. CNET News.com.
- [17] Site web du Java Community Process (JCP). <http://www.jcp.org/>.
- [18] JCP 2 : Process Document. <http://www.jcp.org/en/procedures/jcp2>, March 2004. Version 2.6. Document décrivant le fonctionnement du JCP.
- [19] Executive Committee Info. <http://jcp.org/en/participation/committee>. Composition de l'Executive Committee du JCP.
- [20] Java Specification Participation Agreement. <http://www.jcp.org/aboutJava/communityprocess/JSPA2.pdf>, October 2002. Version actuellement en vigueur.

- [21] JSR 176 : J2SE 5.0 (Tiger) Release Contents. <http://www.jcp.org/en/jsr/detail?id=176>, September 2004.
- [22] JDK 5.0 Documentation. <http://java.sun.com/j2se/1.5.0/docs/index.html>, 2004.
- [23] Specification : Java 2 platform standard edition development kit 5.0 specification ("specification"). <http://java.sun.com/j2se/1.5.0/docs/relnotes/license.html>, August 2004.
- [24] Sonali Shah and Frank Sommers. Does the JCP adequately balance innovation with maintenance of Java's standards? JavaWorld. <http://www.javaworld.com/javaworld/jw-11-2002/jw-1108-jcp.html>, November 2002.
- [25] Frank Sommers and Sonali Shah. A fact-based analysis of the JCP's effectiveness. JavaWorld. <http://www.javaworld.com/javaworld/jw-01-2003/jw-0131-jcp.html>, January 2003.
- [26] JSPA news & status. <http://jakarta.apache.org/site/jspa-position.html>, 2002. La position de la fondation Apache pour la révision du JSPA.
- [27] Robert A. Gingell. Java community process (JCP) program chair responds to apache software foundation. <http://jcp.org/aboutJava/communityprocess/announce/LetterofIntent.html>. La position de Sun pour la révision du JSPA.
- [28] Sun and INRIA kick-off J2EE license agreement for the certification of JOnAS. <http://www.objectweb.org/phorum/read.php?f=25&i=61&t=61>, April 2004.
- [29] Richard Stallman. Free but shackled - the Java trap. <http://www.gnu.org/philosophy/java-trap.html>, April 2004.
- [30] Havoc Pennington. Java licensing. <http://log.ometer.com/2004-04.html#13>, April 2004.
- [31] Jonathan Skillings. Kodak sues sun over aspects of java. <http://news.com.com/2100-1001-836322.html>, February 2002. CNET News.com.
- [32] Ben Rand. Kodak wins Java lawsuit. *Rochester Democrat and Chronicle*, October 2004. <http://www.rochesterdandc.com/apps/pbcs.dll/article?AID=/20041002/BUSINESS/410020333>.
- [33] Rob Malda (a.k.a. CmdrTaco). Kodak wins \$1 billion Java lawsuit. <http://yro.slashdot.org/yro/04/10/03/1352252.shtml>, Octobre 2004.
- [34] CowboyNeal. Sun and Kodak settle out of court. <http://yro.slashdot.org/article.pl?sid=04/10/07/2358245>, Octobre 2004.
- [35] Pamela Jones. Kodak wins Java lawsuit against Sun. <http://www.groklaw.net/article.php?story=20041003041632172>, October 2004.
- [36] Pamela Jones. Sun and kodak settle for \$92 million. <http://www.groklaw.net/article.php?story=20041007164855587>, October 2004.
- [37] Sun Microsystems stands behind its customers and communities, settling outstanding litigation with Eastman Kodak company. <http://www.sun.com/smi/Press/sunflash/2004-10/sunflash.20041007.2.html>, October 2004. Communiqué de presse de Sun sur l'accord avec Kodak.

- [38] Stephen Shankland. Sun settles Kodak's Java suit for \$92 million. http://news.com.com/2100-1012_3-5401804.html, October 2004. CNET News.com.
- [39] Charles Babcock and John Foley. The cost of ideas. <http://www.informationweek.com/story/showArticle.jhtml?articleID=49900578>, October 2004. Information Week.
- [40] Site web de Jikes. <http://jikes.sourceforge.net/>.
- [41] Site web de Classpath. <http://www.gnu.org/software/classpath/>.
- [42] Site web de Jikes RVM. <http://jikesrvm.sourceforge.net/>.
- [43] Site web de Kaffe. <http://www.kaffe.org/>.
- [44] Site web de SableVM. <http://sablevm.org/>.
- [45] Site web de IKVM.NET. <http://www.ikvm.net/>.
- [46] Site web de GCJ. <http://gcc.gnu.org/java/>.
- [47] JBoss, inc. delivers JBoss application server 4.0 to the enterprise market. <http://www.jboss.org/services/press/jboss4.pdf>, September 2004. Annonce de presse de JBoss Inc. pour la certification de JBoss 4.0 en tant que serveur J2EE.
- [48] ECMA and ISO/IEC C# and common language infrastructure standards. <http://msdn.microsoft.com/net/ecma/>, 2003. Description du processus de normalisation de .NET par Microsoft.
- [49] ECMA international. *C# Language Specification*, second edition, December 2002. Standard ECMA-334, disponible à l'URL <http://www.ecma-international.org/publications/standards/Ecma-334.htm>.
- [50] ECMA international. *Common Language Infrastructure (CLI) Partitions I to V*, second edition, December 2002. Standard ECMA-335, disponible à l'URL <http://www.ecma-international.org/publications/standards/Ecma-335.htm>.
- [51] Martin LaMonica. Microsoft to score new C# standard. http://news.com.com/2100-1007_3-995108.html, April 2003.
- [52] *Draft de la spécification de C# 2.0*. <http://www.mondrian-script.org/ecma/>.
- [53] Microsoft. *Bibliothèque de classes .NET Framework*. http://msdn.microsoft.com/library/fre/default.asp?url=/library/FRE/cpref/html/cpref_start.asp.
- [54] David Berlind. Will C# benefit Microsoft, or the industry? <http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2887217,00.html>, October 2002. ZDNet.
- [55] Gabriel Bouvigne. Patents and mp3. <http://www.mp3-tech.org/patents.html>.
- [56] Eric Scheirer. Frequently asked questions : MPEG, patents, and audio coding. <http://web.media.mit.edu/~eds/mpeg-patents-faq>, October 1998.
- [57] Thomson et les licences mp3. <http://www.mp3licensing.com/>.
- [58] Site web du projet ogg. <http://www.xiph.org/ogg/>.
- [59] Code of conduct in patent matters. <http://www.ecma-international.org/memento/codeofconduct.htm>. La politique de l'ECMA en matière de brevets.
- [60] Andrew Orlowski. Ms patents .Everything. http://www.theregister.co.uk/2003/02/11/ms_patents_everything/, February 2003. The Register.

- [61] Lisa M. Bowman. .Net patent could stifle standards effort. http://news.com.com/2100-1001_3-984052.html, February 2003. CNET News.com.
- [62] Jim Miller. Microsoft applies for .Net patent. <http://web.archive.org/web/20030609164123/http://mailserver.di.unipi.it/pipermail/dotnet-sscli/msg00218.html>, February 2003. Déclaration sur le brevet sur l'API de .NET par un des inventeurs du brevet.
- [63] Intellectual property licensing. <http://www.microsoft.com/mscorp/ip/>, August 2004.
- [64] Seth Nickell. Why mono is currently an unacceptable risk. <http://www.gnome.org/~seth/blog/mono>, May 2004.
- [65] Mike Ricciuti. Gates wades into open-source debate. <http://news.com.com/2100-1001-268667.html>, June 2001. CNET News.com.
- [66] John Lettice. GPL Pacman will eat your business, warns Gates. http://www.theregister.co.uk/2001/06/20/gpl_pacman_will_eat_your/, June 2001. The Register.
- [67] Shared source initiative : Licensing overview. <http://www.microsoft.com/resources/sharedsource/Articles/LicensingOverview.aspx>, February 2004. Une présentation des licences libres par Microsoft.
- [68] Présentation du Sender ID sur la Wikipedia. http://en.wikipedia.org/wiki/Sender_id.
- [69] Standards licensing programs. <http://www.microsoft.com/mscorp/ip/standards/>. Les licences des standards Microsoft, dont celle pour le sender ID.
- [70] ASF position regarding Sender ID. <http://apache.org/foundation/docs/sender-id-position.html>, September 2004.
- [71] Le projet Debian rejette l'utilisation de Sender ID. <http://www.debian.org/News/2004/20040904>, Septembre 2004.
- [72] Site web de l'Internet Engineering Task Force (IETF). <http://www.ietf.org/>.
- [73] Sendmail, Inc. position on Sender ID. <http://www.sendmail.net/sid-milter/smi-position.shtml>, September 2004.
- [74] Yakov Shafranovich. Sender-ID back from the dead. http://www.circleid.com/article/792_0_1_0_C/, October 2004. Circle ID.
- [75] Dossier sur le reverse engineering. MISC Numéro 14, Juillet-Août 2004.
- [76] CIFS Sub-Group of the SNIA NAS TWG. *Common Internet File System Technical Reference*. Storage Networking Industry Association, February 2002. http://www.snia.org/tech_activities/CIFS/.
- [77] CIFS, Microsoft and the Samba Team. http://www.samba.org/samba/ms_license.html.
- [78] Stephen Shankland. Microsoft steps on Samba's toes. http://news.zdnet.com/2100-3513_22-904089.html, May 2002. ZDNet News.
- [79] Royalty-free CIFS technical reference license agreement. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cifs/protocol/royalty-free_cifs_technical_reference_license_agreement.asp. Licence Microsoft pour la documentation de CIFS.

- [80] Microsoft bars GNU software from interoperating with CIFS. <http://swpat.ffii.org/patents/effects/cifs/>.
- [81] Microsoft and patents. <http://swpat.ffii.org/players/microsoft/>.
- [82] Site web de VirtualDub. <http://www.virtualdub.org/>.
- [83] Andy Tai. Microsoft patents ASF media file format, stops reverse engineering. <http://www.advogato.org/article/101.html>, June 2000. Advogato.
- [84] Advanced systems format (ASF) specification . <http://www.microsoft.com/windows/windowsmedia/format/asfspec.aspx>. Spécifications du format ASF.
- [85] Stephen Shankland. HP memo : Microsoft planned open-source patent fight. http://news.com.com/2100-7344_3-5276901.html, July 2004. CNET News.com.
- [86] Ina Fried. Microsoft opens technology to more licensing. http://news.com.com/2100-1012_3-5113033.html, December 2003. CNET News.com.
- [87] Antone Gonsalves. Microsoft's patent plans worry open-source supporters. <http://www.techweb.com/wire/story/TWB20040604S0005>, June 2004. TechWeb.com.
- [88] FAT file system technology and patent license. <http://www.microsoft.com/mscorp/ip/tech/fat.asp>, December 2003. Dernière mise à jour en mars 2005.
- [89] Ina Fried. Un brevet de microsoft sur le système FAT déclaré invalide. <http://www.zdnet.fr/actualites/informatique/0,39040745,39175892,00.htm>, Octobre 2004. ZDNet.fr.
- [90] C.C./IDG News Service. Microsoft ambigu avec les développeurs Open Office. http://www.web1mi.com/news_store/2004_09_16_Microsoft_ambigu_ave_63/News_view, Septembre 2004. Le Monde Informatique.
- [91] Site web du projet Mono. <http://www.mono-project.com/>.
- [92] Martin LaMonica. More than an open-source curiosity. http://news.com.com/2008-7344_3-5271084.html, July 2004. CNET News.com.
- [93] Miguel de Icaza. On software patents. <http://tirania.org/blog//texts/patents.html>, March 2004.
- [94] Miguel de Icaza. More patents. <http://tirania.org/blog/archive/2004/May-20.html>, May 2004.
- [95] FAQ : Licensing. http://www.mono-project.com/FAQ:_Licensing.
- [96] Blog de nat friedman, co-fondateur de ximian avec de icaza. <http://nat.org/2004/may/#21-May-2004>.
- [97] Blog de paolo morano. <http://www.advogato.org/person/lupus/diary.html?start=11>.
- [98] Site web du projet DotGNU. <http://www.dotgnu.org/>.
- [99] Informations sur les brevets logiciels en Europe. <http://swpat.ffii.org/>.

A Licence

Cette création est mise à disposition selon le Contrat Paternité - Pas d'Utilisation Commerciale - Pas de Modification disponible en ligne <http://creativecommons.org/licenses/by-nc-nd/2.0/fr/> ou par courrier postal à Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

B Historique

21 Juillet 2005 Passage en deux colonnes pour le texte principal

16 Juin 2005 Mise en ligne avec les modifications suivantes :

- passage en LaTeX
- correction de quelques fautes de frappe
- ajout des conditions de distribution et de l'historique
- unification des références
- vérification et correction de tous les liens

Décembre 2004 Publication par le GNU/Linux Magazine France