

# Java, .NET et les logiciels libres

## 2.0

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

7 septembre 2006

### Résumé

Les logiciels libres construits à partir des plateformes Java et .NET ont de plus en plus de succès. De l’environnement de développement Eclipse aux lecteurs de flux RSSOwl et Blam, en passant par le moteur de recherche locale Beagle et le client BitTorrent Azureus, on constate que de nombreux développeurs se détournent des langages plus classiques comme le C ou le C++. Même en occultant l’important débat technique, notamment sur la consommation en ressources (temps cpu et mémoire) des plateformes Java et .NET, on sait que choisir une de ces technologies n’est pas sans conséquence importante : Java et .NET ne sont que partiellement ouverts, ce qui peut poser des problèmes de pérennité pour les logiciels libres développés en les utilisant.

Suite aux annonces importantes de l’année 2006 (Mono dans Fedora Core 5 et dans Gnome 2.16, le JDK de Sun bientôt libre), cet article fait le point sur la situation légale des plateformes Java et .NET et tente d’évaluer leur compatibilité avec le logiciel libre.

### 1 Introduction

Il y a bientôt deux ans, je présentais dans [1] la situation légale très complexe des plateformes de développement Java et .NET. Au delà de leurs ressemblances techniques (machine virtuelle, compilation *just in time*, gestion automatique de la mémoire, langage orienté objet inspiré du C++, etc.), ces plateformes ont aussi en commun d’avoir été conçues par des entreprises et d’avoir comme implémentation de référence un logiciel propriétaire. En fait, Java et .NET sont des technologies partiellement ouvertes.

Les spécifications de la plateforme Java autorisent explicitement une implémentation *open source*<sup>1</sup>, mais sous des conditions assez restrictives. L’implémentation de référence distribuée par Sun contient des éléments libres, mais est globalement un logiciel propriétaire.

La plateforme .NET est standardisée par l’ECMA [3] et par l’ISO [4] (cf les standards ECMA-334 [5] et ECMA-335 [6]). Selon les règles de ces organismes (cf [7]), les standards qu’ils produisent peuvent être implémentés par quiconque. Si les technologies correspondantes sont protégées par des

---

<sup>1</sup>J’utilise dans cet article “logiciel *open source*” comme synonyme de “logiciel libre”, en me référant à la définition de l’*open source* donnée par l’OSI [2].

brevets, ceux-ci doivent être licenciés en des termes “raisonnables et non discriminatoires” (RAND pour Reasonable And Non Discriminatory). L’implémentation de référence de .NET est un logiciel propriétaire de Microsoft.

De ce fait, et contrairement à des langages conçus de façon complètement ouverte et à l’implémentation de référence libre (comme Python, Perl ou Ruby, par exemple), ou à des langages standardisés et implémentables sans restriction aucune (comme C, C++ ou Ada), .NET et Java ne sont que partiellement ouverts, tant au niveau de la conception (qui passe par des comités à l’accès restreint) que de l’implémentation. Un développeur de logiciels libres doit donc se poser la question des risques qu’il prend en utilisant ces plateformes (cf par exemple le texte de Richard Stallman sur Java [8]). En effet, quand on développe une application Python, par exemple, on a la garantie qu’on pourra toujours distribuer librement (au sens des logiciels libres) cette application avec un environnement Python, ce qui assure sa pérennité en tant que logiciel *open source*. Une application C n’utilisant que des bibliothèques libres offre les mêmes garanties. C’est le cas plus généralement de la grande majorité des langages de programmation et des plateformes de développement.

La situation d’une application développée en Java ou .NET est moins claire. Tout d’abord, il n’existe aucune implémentation libre complète de ces plateformes<sup>2</sup> : en utilisant certaines fonctionnalités avancées de la plateforme, une application libre ne peut pas être exécutée de façon “libre”. De plus, et c’est bien plus grave, le caractère propriétaire des technologies elles-mêmes (et pas seulement de leurs implémentations de référence) menace l’implémentation libre des

plate-formes, essentiellement en raison des brevets couvrant ces dernières.

Les éléments présentés dans mon précédent article [1] m’amenaient à insister sur les risques inhérents aux deux plateformes. Dans le cas de Java, je constatais qu’il était théoriquement possible de réaliser une implémentation *open source* de la plateforme, mais en respectant des conditions de conformité aux spécifications extrêmement restrictives, ce qui présentait un risque important : le développement ouvert des logiciels libres implique une distribution de versions incomplètes d’une implémentation *open source* de Java, et donc un non respect (au moins temporaire) des conditions de conformité.

Concernant .NET, j’étais encore plus négatif, essentiellement en raison d’une demande de brevet déposée par Microsoft et portant sur une grande partie de l’API de la plateforme. Une fois cette demande acceptée, Microsoft aurait obtenu le contrôle total des implémentations de .NET et aurait pu se contenter de proposer des licences RAND incompatibles avec une implémentation libre (quelles soient *Royalty Free* ou non).

Depuis décembre 2004, la situation a évolué, tant au niveau des implémentations libres qui ont beaucoup progressé, que du point de vue légal. Le but du présent article est d’analyser la situation actuelle et ses conséquences pour l’utilisation des plateformes Java et .NET pour le développement de logiciels libres.

## 2 Logiciels libres et brevets

Avant de s’intéresser à Java et .NET, il est intéressant de rappeler quelques éléments sur les brevets logiciels en général. D’autres aspects seront abordés lors de l’analyse de brevets portant sur .NET.

---

<sup>2</sup>début septembre 2006.

## 2.1 Brevets “spécifiques” et “généraux”

Quand on développe un logiciel, on s'expose malheureusement au risque de contrefaçon d'une invention protégée par un brevet. Certaines systèmes légaux, en particulier celui des États-Unis d'Amérique, autorisent en effet de breveter des algorithmes en considérant que ceux-ci sont réalisés dans un programme au même titre qu'une idée est réalisée dans une construction matérielle brevetée. Ces brevets logiciels interdisent dans les faits d'implémenter l'algorithme protégé (ou une amélioration de celui-ci), quel que soit le langage de programmation et la licence du logiciel. Le droit d'implémentation s'obtient sous la forme d'une licence d'implémentation, délivrée par le propriétaire du brevet.

Quand on implémente un standard, ou, plus généralement, quand on cherche à reproduire les fonctionnalités d'un logiciel déjà existant, il faut considérer deux types de brevets :

1. les brevets déposés par les inventeurs du standard (ou du logiciel modèle) et qui visent explicitement à protéger certains aspects de celui-ci ;
2. les brevets déposés par des tiers et qui s'appliquent à l'implémentation en cours de développement.

Les brevets de la première catégorie, qu'on peut considérer comme “spécifiques”, sont généralement mis en avant par les inventeurs du standard ou du logiciel, ce qui permet d'évaluer les risques de contrefaçon avant même de commencer le développement. Le problème avec ce type de brevets est qu'ils ont souvent été déposés pour contrôler au maximum les possibilités de concurrence et qu'il est donc parfois totalement impossible de les contourner. Le standard MP3,

par exemple, est couvert par des brevets détenus par le Fraunhofer Institute et Thomson Multimedia (cf [9, 10], ainsi que [11] pour des précisions). Il semble impossible de développer autour de MP3 (lecteur ou encodeur) sans mettre en œuvre les brevets (et donc sans obtenir une licence de Thomson), ce qui explique<sup>3</sup> l'absence de support de ce format sonore dans Fedora Core [12], par exemple.

Contrairement aux problèmes posés par des brevets spécifiques, ceux liés à des brevets de la deuxième catégorie (qu'on peut qualifier que “brevets généraux”) sont très difficile d'anticiper . Ces brevets peuvent toucher n'importe quel logiciel libre, depuis le “bas niveau” du noyau Linux et du compilateur gcc, jusqu'au “haut niveau” des applications utilisateurs classiques comme OpenOffice.org. Il est parfois possible de contourner de tels brevets, soit parce qu'ils couvrent un élément qui n'est pas crucial à l'application (par exemple la gestion, dans un logiciel comme Gimp, du format gif quand celui-ci était encore breveté par Unisys) et qu'il suffit donc de supprimer cet élément, soit encore car il existe une façon d'obtenir la même forme de résultat sans pour autant implémenter l'algorithme breveté.

Dans la suite de cet article, je m'intéresse avant tout aux brevets de la première catégorie, car ils posent des problèmes spécifiques aux plateformes que j'étudie, puisqu'ils ont été conçus pour protéger celles-ci et s'appliqueront donc à des implémentations libres de façon quasi-automatique. Ils mettent en péril ces implémentations, et donc touchent indirectement tout logiciel libre développé pour une

---

<sup>3</sup>Pour être précis, les conditions de redistribution imposées par la licence qu'on peut obtenir de Thomson sont incompatibles avec une implémentation *open source*, ce qui motive la décision de Red Hat.

de ces plateformes. Je mentionnerai cependant aussi des brevets généraux, car certains ont été divulgués.

## 2.2 Licence RAND

Pour certains organismes de standardisation comme l'ISO et l'ECMA, une licence RAND est caractérisée par des termes "raisonnables et non discriminatoires" (traduction de *Reasonable And Non Discriminatory*). Ces organismes exigent de leurs membres que les brevets nécessaires à l'implémentation d'un standard soient licenciés en termes RAND.

Il n'existe pas de définition formelle des termes RAND<sup>4</sup>, mais l'aspect important semble être le caractère non discriminatoire : les termes de la licence doivent dépendre de l'utilisation du brevet et non pas de l'utilisateur. La redevance à acquitter doit aussi être "raisonnables", c'est-à-dire compatible avec une utilisation commerciale de l'implémentation. A titre d'exemple, on peut consulter les tarifs de Thomson pour le standard MP3 [14] : si les développeurs de LAME [15] devaient respecter les brevets de Thomson, ils auraient à payer entre 2,5 et 5 US \$ par copie du logiciel (et ils devraient bien entendu changer leur licence, incompatible avec celle des brevets MP3).

Je ne reviendrais pas sur l'incompatibilité générale entre les licences RAND pour les brevets et les licences libres pour les logiciels. J'ai longuement abordé ce point dans mon précédent article et je renvoie le lecteur à celui-ci pour les détails (cf [1], en particulier la section 4.4). Notons que ce n'est pas uniquement l'aspect financier qui pose problème dans le RAND, car une licence

---

<sup>4</sup>La Free Software Foundation propose de remplacer RAND par un acronyme plus explicite, UFO, pour *Uniform Fee Only*, c'est-à-dire licence à redevance uniforme [13].

RAND peut parfaitement ne demander aucune redevance. Le problème est plutôt lié aux conditions de redistribution et de sous-licence (cf l'analyse de Seth Nickell [16] ou encore celle de la fondation Apache [17], ainsi que les déclarations de Michele Herman, de Microsoft, rapportées dans [18]).

## 3 .NET et les brevets logiciels

### 3.1 Rappels sur la plateforme

Rappelons tout d'abord que .NET est une plateforme *partiellement* standardisée. Le cœur de la technologie est la spécification appelée *Common Language Infrastructure* qui décrit une machine virtuelle d'assez haut niveau. La CLI est entièrement couverte par le standard ECMA-335 [6], qui spécifie aussi des bibliothèques de base, ainsi que des profils, c'est-à-dire des jeux de bibliothèques. L'autre élément important de .NET est son langage de prédilection, le C# [19], qui est entièrement spécifié par le standard ECMA-334 [5].

En tant que standard ISO et ECMA, ces deux éléments fondamentaux de la plateforme .NET sont donc implémentables librement, sous réserve de l'obtention d'une licence (au pire RAND [7]) pour les éventuels brevets couvrant les technologies mise en œuvre pour l'implémentation.

A ce cœur de plateforme s'ajoute de nombreux bibliothèques développées par Microsoft, comme par exemple Windows.Forms qui permet de créer des interfaces graphiques ou encore ADO.NET qui regroupe des API d'accès à des bases de données. Ces bibliothèques, qui représentent l'essentiel de .NET<sup>5</sup>, ne sont pas standardisées par

---

<sup>5</sup>La version 1.0 de .NET contenait environ 3 800 classes dont seulement 350 étaient standardisées [18].

l'ECMA et ne sont donc pas couvertes par sa politique de brevets [7].

### 3.2 Des brevets sur .NET ?

De prime abord, il semble difficile de savoir si les inventeurs de .NET (Microsoft en majorité, mais aussi Hewlett-Packard et Intel) possèdent des brevets spécifiques qui couvrent des éléments cruciaux de la partie standardisée de la plateforme, car les déclarations des intéressés sont relativement vagues.

Lors du CeBIT, en mars 2002, Steve Ballmer (PDG de Microsoft), interrogé au sujet d'un soutien éventuel aux implémentations alternatives de .NET comme Mono, répondait [20]

*“Wir haben so viele Millionen in .NET gesteckt, wir haben so viele Patente auf .NET, die wir pflegen wollen<sup>6</sup>.”*

Dans un entretien accordé à David Berlind en 2002 [18], Michele Herman, responsable de la “propriété intellectuelle” chez Microsoft, précisait

*we believe we have a patent or one pending that's essential to implementing the specification<sup>7</sup>.*

En 2002, Microsoft annonçait donc posséder au plus un brevet indispensable à l'implémentation des deux standards ECMA. Un des inventeurs de .NET, Jim Miller, était tout aussi évasif en février 2003 sur une liste de diffusion concernant la plateforme [21]

<sup>6</sup>“Nous avons investis tant de millions dans .NET et nous avons obtenus tant de brevets sur .NET, que nous voulons les faire fructifier.”

<sup>7</sup>“nous pensons avoir un brevet ou une demande de brevet essentiel à l'implémentation de la spécification.”

*But Microsoft (and our co-sponsors, Intel and Hewlett-Packard) went further and have agreed that our patents essential to implementing C# and CLI will be available on a “royalty-free and otherwise RAND” basis for this purpose<sup>8</sup>.*

Miller précise la nature des licences qui seront accordées pour les brevets essentiels, mais n'indique pas s'il existe de tels brevets.

En juillet 2004, Miguel de Icaza, leader du projet Mono [22], implémentation libre de .NET financée par Novell, indiquait dans un entretien donné à Martin Lamonica [23]

*Well, at this point, we don't know of any patent infringement that Mono has (committed)<sup>9</sup>.*

De Icaza confirmait ici ce qu'il avait déjà indiqué lors de la rencontre des développeurs Mono de mars 2004, à savoir que Novell avait réalisé en 2004 une étude d'antériorité pour déterminer si leur implémentation de .NET mettait en œuvre des algorithmes brevetés [24] et avait conclu à l'absence de contrefaçon.

En recoupant les déclarations des employés de Microsoft et les affirmations de De Icaza, on pourrait croire que Microsoft ne possède finalement qu'une demande de brevet concernant directement .NET (ou au moins la partie standardisée de la plateforme). Comme nous allons le voir dans la suite de l'article, cette vision idyllique est malheureusement démentie par les faits.

<sup>8</sup>“Mais Microsoft (et nos autres sponsors, Intel et Hewlett-Packard) sont allés plus loin et ont donné leur accord pour que nos brevets essentiels à l'implémentation de C# et de la CLI soit licencié selon des termes “sans redevance ainsi que RAND” pour cet but [l'implémentation des standards].”

<sup>9</sup>“Et bien pour l'instant, nous n'avons pas connaissance d'une contrefaçon de brevet que Mono pourrait commettre.”

### 3.2.1 Une demande de brevet sur l'API de .NET

Une demande de brevet [25] portant sur l'API de .NET a été faite en février 2002 et rendue publique un an plus tard, en février 2003. Elle a beaucoup attiré l'attention de la presse en ligne [26, 27] et plus généralement des personnes s'intéressant à .NET. C'est elle qui a d'ailleurs déclenché la fameuse déclaration de Jim Miller sur les licences (citée plus haut).

Cette demande est caractéristique des dérives du système de brevets américains, puisqu'elle porte sur une API, c'est-à-dire sur une liste de spécifications de types abstraits et de fonctions! Le résumé de la demande est édifiant

*“An application program interface (API) provides a set of functions for application developers who build Web applications on a network platform, such as Microsoft Corporation's .NET™ platform<sup>10</sup>.”*

En d'autres termes, la demande de brevets prétend protéger une liste de fonctions pour développer des applications réparties. Cette liste contient les bibliothèques de base de .NET (spécifiées dans le standard ECMA-335 [6]), ainsi que de nombreuses autres bibliothèques (en particulier ADO.NET, les Windows.Forms, etc.). La partie descriptive de la demande présente la CLI (dont la spécification ECMA est annexée à la demande de brevet), ainsi que ses principes généraux (en particulier l'aspect multilingage). Elle décrit ensuite de façon assez détaillée les API annexées à la demande.

Il est important cependant de noter qu'un

---

<sup>10</sup>“Une interface de programmation qui fournit un ensemble de fonctions pour les développeurs qui construisent des applications Web sur une plateforme répartie, comme la plateforme .NET de Microsoft.”

brevet américain ne couvre que les revendications (les *claims*) et pas la partie descriptive. Pour comprendre la portée de la demande de brevet de Microsoft, il faut donc analyser les 41 revendications de celle-ci. Or, la première revendication est la suivante

*“A software architecture for a distributed computing system comprising :*

*[...]*

*a common language runtime layer that can translate Web applications written in different languages into an intermediate language supported by the common runtime layer<sup>11</sup>.”*

Cette revendication couvre donc toute architecture logicielle qui passe par un langage intermédiaire pour son exécution, à condition que le système supporte plusieurs langages et qu'il soit destiné à la production d'applications Web distribuées. Les autres revendications décrivent de façon de plus en plus détaillée les API annexées à la demande. Les descriptions restent cependant très vagues et donc très générales. Elles peuvent donc couvrir l'API de .NET, mais aussi plus généralement des API assez similaires dans leurs principes. Par exemple la cinquième revendication est formulée comme suit

*“An application program interface embodied on one or more computer readable media, comprising :*

*a first group of services related to creating Web applications ;*

*a second group of services related*

---

<sup>11</sup>“Une architecture logicielle pour un système informatique réparti comprenant : [...] une couche d'exécution pour un langage commun capable de traduire des applications Web écrites dans divers langages dans un langage intermédiaire compatible avec la couche d'exécution commune.”

to constructing client applications ;  
 a third group of services related  
 to data and handling XML docu-  
 ments ; and  
 a fourth group of services related  
 to base class libraries ; and further  
 comprising :  
 a common language runtime  
 layer that can translate Web ap-  
 plications written in different lan-  
 guages into an intermediate lan-  
 guage supported by the common  
 runtime layer<sup>12</sup>.”

Face à des revendications aussi générales, on ne voit pas comment une implémentation libre de .NET comme Mono pourrait ne pas enfreindre le brevet Microsoft, s’il était accordé. Une manière d’échapper au brevet consisterait cependant à éviter de satisfaire l’ensemble des closes des revendications. En effet, la cinquième revendication (par exemple) porte sur une API qui inclut les cinq éléments listés. Il “suffit” donc de supprimer de l’API le groupe permettant la création d’application web (c’est-à-dire essentiellement ASP.NET) ou encore celle permettant la création d’applications clientes (en gros Windows.Forms) pour ne pas enfreindre la revendication<sup>13</sup>.

<sup>12</sup>“Une interface de programmation, inscrite sur un ou plusieurs support(s) lisible(s) par un ordinateur, comprenant : un premier groupe de services associés à la création d’applications Web ; un deuxième groupe de services associés à la création d’applications clientes ; un troisième groupe de services associés à la gestion de données et de documents XML ; un quatrième groupe de services associés à une bibliothèque de classes de bases ; et comprenant de plus une couche d’exécution pour un langage commun capable de traduire des applications Web écrites dans divers langages dans un langage intermédiaire compatible avec la couche d’exécution commune.”

<sup>13</sup>À moins, bien entendu, qu’un autre brevet couvre une combinaison moins large, ce qui est tout à fait possible. On quitte cependant le cas spécifique

La bonne nouvelle est que le brevet ne semble pas sur la bonne voie (du point de vue de ses inventeurs). Le bureau des brevets américains à en effet notifié Microsoft de sa décision de rejeter la demande (le 4 mai 2006). L’entreprise dispose de trois mois pour apporter des arguments et des modifications pour que la demande soit de nouveau examinée en appel (cette période peut être étendue à six mois). Nous saurons donc dans peu de temps si Microsoft tente de nouveau sa chance avec cette API ou si l’entreprise préfère abandonner la demande<sup>14</sup>.

La mauvaise nouvelle réside dans les arguments utilisés par le bureau des brevets. En fait, les revendications sont toutes rejetées car elles sont considérées comme des conséquences triviales d’autres revendications contenue dans des brevets accordés ou dans des demandes. Plus précisément :

- d’une part, l’examineur de la demande de Microsoft considère qu’un brevet de Roberts et al. [28] couvre les systèmes répartis proposant d’un côté des services web et de l’autre une API associée ;
- d’autre part, il précise qu’une demande de David Yach [29] contient l’idée d’utiliser un langage intermédiaire (le fameux *common language runtime layer* de la demande de Microsoft) pour traduire des applications Web écrites dans d’autres langages ;
- enfin, il considère que la combinaison des deux est triviale et donc non brevetable, ce qui invalide une grande par-

de .NET pour retomber dans le problème général des brevets logiciels.

<sup>14</sup>Notons qu’un abandon ne signifie pas une mise à la poubelle de la demande ou encore un passage dans le domaine public de son contenu : le mécanisme des “continuations” permet à un inventeur de déposer une nouvelle demande qui reprend les éléments révélés dans la précédente.

tie des revendications de la demande de Microsoft.

L'examinateur invoque aussi un autre brevet de Microsoft [30] dont les éléments (en particulier tout ce qui concerne la gestion des fichiers de configuration) peuvent aussi être combinés de façon simple avec ceux de [28] pour invalider les autres revendications.

Il est difficile de bien évaluer les conséquences d'un rejet de la demande de brevet de Microsoft. Si le bureau des brevets maintient ses arguments, .NET lui-même, dans son incarnation orientée services web, serait finalement une contrefaçon des brevets et demande évoqués ci-dessus. Ce serait vraisemblablement le cas aussi pour une implémentation libre de .NET. En pratique, les inventeurs du brevet [28] et celui de la demande (si elle est accordée) pourraient donc suivre les conclusions du bureau des brevets et attaquer Microsoft ou tout autre auteur d'une implémentation de .NET, sans avoir à suivre la politique de licence de l'ECMA, puisqu'ils n'ont pas participé au processus de standardisation. Il s'agit donc d'un cas de brevets généraux que je mentionnais plus haut.

### 3.2.2 Des brevets validés ou en cours de validation

Mais il y a en fait beaucoup plus gênant que cette demande de brevet qui pourrait bien avoir joué dans la blogosphère le rôle de l'arbre qui cache la forêt<sup>15</sup>. Microsoft a en effet déposé en juin et juillet 2001 six demandes de brevets concernant l'API .NET et deux demandes concernant d'autres aspects de la plateforme. L'une des demandes est devenue celle que j'ai étudiée dans la section

<sup>15</sup>A ce sujet, il est édifiant de ne voir mentionnée que cette demande de brevet dans des articles récents comme par exemple [31], datant de juin 2006.

précédente. Les autres ont eu généralement plus de succès :

1. la demande 09/902,811 est devenue le brevet 7,017,162 (délivré en mars 2006). Son résumé est légèrement différent de celui de la demande précédemment étudiée puisqu'il dit

*“An application program interface (API) provides a set of functions, including a set of base classes and types that are used in substantially all applications accessing the API, for application developers who build Web applications on Microsoft Corporation's .NET™ platform<sup>16</sup>.”*

En fait, le brevet protège un modèle particulier d'appel de méthodes à distance (utilisé par .NET), ainsi que des systèmes qui incluent ce modèle.

2. la demande 09/902,809 est en cours d'acceptation. Elle porte essentiellement sur une API de gestion du protocole HTTP.
3. la demande 09/902,560 est devenue le brevet 6,920,461 (délivré en juin 2006). Le brevet porte essentiellement sur l'API d'interface entre .NET et les bases de données (ADO.NET).
4. la demande 09/902,810 est en cours d'acceptation. Elle porte sur l'API de manipulation des documents au format XML.
5. la demande 09/902,812 est en cours d'appel après rejet par le bureau des

<sup>16</sup>*“Une interface de programmation qui fournit un ensemble de fonctions, incluant un ensemble de classes et de types de bases qui sont utilisés dans la quasi totalité des applications utilisant l'interface, pour les développeurs qui construisent des applications Web basées sur la plateforme .NET de Microsoft.”*

brevets. Elle porte sur une API de construction de clients graphiques pour des systèmes répartis. Les arguments du rejet sont intéressants car ils font pour une fois référence à l'art antérieur en dehors du système des brevets, par exemple en utilisant un simple manuel Java!

6. la demande 09/598,105 est devenue le brevet 6,836,883 (délivré en décembre 2004). Il s'intitule "*Method and system for compiling multiple languages*<sup>17</sup>" et porte sur le cœur de .NET, la CLI. Le brevet protège un système dans lequel on compile un fichier source vers un code exécutable (éventuellement pour une machine virtuelle) et simultanément vers un fichier de métadonnées qui décrit les fonctionnalités proposées par ce source (en gros, l'interface des fonctions et les types définis). Comme ce genre de système est breveté depuis longtemps, l'examineur du brevet a fait réécrire les revendications d'une façon relativement complexe dans laquelle le système doit nécessairement traiter au moins deux langages différents pour être protégé.

7. la demande 09/613,289 a été abandonnée dans sa formulation de base, mais est poursuivie par deux demandes :

(a) la demande 09/614,158 est devenue le brevet 6,738,968 (délivré en mai 2004). Il s'intitule *Unified data type system and method* et porte sur le système de représentation double de certains types dans la CLI : la version native (par exemple un entier représenté par

un int codé sur 4 octets) et la version *boxed* dans laquelle la valeur est encapsulée dans un objet.

(b) la demande 10/848,402 est en cours d'examen et porte aussi sur ce système de représentation double.

En résumé, Microsoft possède donc au moins deux brevets portant sur l'API de .NET et deux autres brevets portant sur le cœur de la CLI (compilation et système de type). Deux autres brevets sur l'API devraient être acceptés dans les mois qui viennent. En outre, quand on recherche les poursuites de demande ou les citations de brevets par d'autres brevets, on constate que Microsoft possède de nombreux autres brevets (ou demandes en cours) autour de .NET. Par exemple, le brevet 6,738,968 sur le système de type de .NET est cité par trois autres brevets de Microsoft portant respectivement sur du *mapping* relationnel, de la conversion automatique de code entre différents langages et de la transformation automatique de hiérarchies d'objets (toujours entre langages différents).

Une autre façon de trouver des brevets concernant .NET consiste simplement à chercher ceux dont l'un des auteurs est Anders Hejlsberg, l'architecte de .NET. Outre les brevets déjà cités, on trouve trois autres brevets portant respectivement sur

1. l'aspect réparti de l'API de .NET (brevet numéro 7,013,469, délivré en mars 2006 et dans le même esprit que le brevet 7,017,162) ;
2. les méthodes de gestion de version utilisées dans .NET (brevet numéro 6,981,250, décembre 2005) ;
3. le système de gestion des événements (basé sur la notion de *delegates*) utilisé dans .NET (brevet numéro 6,951,022,

---

<sup>17</sup> "*Méthode et système pour compiler plusieurs langages*"

septembre 2005).

On trouve aussi 15 demandes de brevets impliquant Hejlsberg.

### 3.2.3 Couverture des brevets

La lecture des différents brevets de Microsoft sur .NET permet de dégager quelques tendances. Les brevets ne pouvant protéger que des innovations, Microsoft a naturellement chercher à faire ressortir dans ses demandes les parties originales de .NET.

Le fait que la CLI ait été explicitement conçue pour gérer plusieurs langages a été par exemple très utilisé, en particulier dans le brevet 6,836,883. Cependant, la solution technique choisie, qui consiste à passer par une description générique des services (fonctions et types) proposés par un module est tellement peu originale que Microsoft a été obligé de faire apparaître explicitement dans ses revendications le fait que le système gérait plusieurs langages. La portée du brevet correspondant est donc assez limitée et il me semble qu'une implémentation de .NET limitée au C# (par exemple) ne serait pas couverte par le brevet en question et plus généralement par tout ce qui concerne la gestion de plusieurs langages.

Un autre aspect que Microsoft a cherché à mettre en avant comme novateur est l'orientation Web, et plus généralement l'aspect réparti de .NET. Le domaine étant de nouveau très concurrentiel, les brevets et les demandes emploient des formulations très spécifiques. Ceci a pour effet de limiter le périmètre des brevets correspondants, mais aussi de rendre délicate l'implémentation d'un "clone" de .NET. Les brevets 7,013,469 et 7,017,162 sur l'API pour les appels à distance sont par exemple très précis (ils indiquent le nom des classes et leur rôle) ce qui rend impossible l'implémentation de l'API

correspondante sans contrefaçon.

Enfin, Microsoft a choisi de protéger des "détails" importants, comme par exemple le système de *boxing* automatique (brevet 6,738,968), la gestion des événements par *delegates* (brevet 6,951,022), ou encore la gestion de version (brevet 6,981,250).

Globalement, Microsoft a donc cherché à verrouiller .NET afin de contrôler toute implémentation du standard et surtout de ces extensions (comme ADO.NET, ASP.NET ou les Windows.Forms). Il faudrait être un expert de l'interprétation des brevets américains pour véritablement mesurer la portée des revendications contenues dans les brevets accordés. Mon point de vue d'amateur est que les verrous ont été bien conçus et qu'il est donc impossible de les contourner car ils portent sur des concepts cruciaux, comme le multi-langages ou le *boxing*. Il est sûrement envisageable de produire des versions restreintes de .NET qui n'enfreignent par certains brevets, par exemple en se limitant au C# ou en évitant certaines bibliothèques, mais un clone de .NET (but du projet Mono, par exemple) semble vouer à contrefaire les produits de Microsoft. On peut se demander en outre si les *bindings* vers des bibliothèques écrites en C (par exemple Gtk#, cf la section suivante), qui sont indispensables pour récupérer l'existant, ne correspondent pas un support multi-langages et n'induisent pas ainsi une contrefaçon cet aspect crucial des brevets de Microsoft.

## 4 Le cas de Mono

### 4.1 Où en est Mono ?

Mono [22] est l'implémentation libre la plus avancée de .NET (la principale alternative, Portable.NET, réalisée par le projet

DotGnu [32], est moins complète et semble progresser moins vite). Mono est dirigé par Miguel de Icaza et sponsorisé par son employeur, Novell. Mono comporte plusieurs éléments :

- une implémentation des deux standards ECMA, c'est-à-dire de la machine virtuelle .NET (CLI), des bibliothèques de base et d'un compilateur C# ;
- un ensemble de bibliothèques de compatibilité avec la version de .NET complète distribuée par Microsoft (ADO.NET, ASP.NET, Windows.Forms, etc.) ;
- un ensemble d'autres bibliothèques originales comme Gtk# (*binding* Gtk pour .NET) et le *Tao Framework* (*binding* OpenGL, SDL, etc.).

La plateforme .NET évolue régulièrement depuis son lancement officiel en janvier 2002 (version 1.0). L'état d'avancement de Mono dans son implémentation de la plateforme doit donc être jugé relativement aux différentes versions de celle-ci (1.1 en sortie en avril 2003 et 2.0 sortie novembre 2005).

Quand elle est sortie en juin 2004<sup>18</sup>, la version 1.0 de Mono proposait une implémentation complète des standards ECMA (dans leur première version), mais une compatibilité seulement partielle avec .NET 1.1, en particulier en raison d'une implémentation très incomplète des Windows.Forms. La prochaine version majeure de Mono (1.2), prévue pour les mois qui viennent<sup>19</sup> doit inclure un support complet des Windows.Forms de .NET 1.1.

Le passage de la version 1.1 de .NET à la version 2.0 a été l'occasion de modifica-

tions importantes au niveau du cœur de la plateforme, en particulier celles induites par l'introduction de la généricité. Le support de ces fonctionnalités est en cours d'intégration dans Mono, mais la version 1.2 ne cherchera pas à être compatible avec .NET 2.0 (certains éléments, comme par exemple le compilateur C# 2.0, sont complets, d'autres sont moins avancés).

Globalement, Mono est donc encore éloigné d'une implémentation complète de la plateforme .NET, même dans sa version 1.1. Malgré cette incompatibilité partielle avec le .NET de Microsoft, Mono propose une plateforme complète de développement multi OS, en particulier grâce à Gtk# qui permet de programmer des interfaces graphiques portables. Il offre aussi un support multi-langages (Java, Visual Basic, etc.).

Pour minimiser les risques liés aux brevets de Microsoft, je pense qu'il est préférable d'utiliser les bibliothèques spécifiques Mono comme Gtk# plutôt que les bibliothèques de compatibilités avec la version Microsoft de .NET comme Windows.Forms. C'est d'ailleurs cette politique qui est suivie par les développeurs des applications Mono les plus connues, en particulier le système d'indexation et de recherche sur le poste client Beagle [33], le gestionnaire de photos numériques F-Spot [34] et l'outil de prise de notes Tomboy [35].

## 4.2 L'équilibre de la terreur

Nous avons vu que Microsoft possède de nombreux brevets touchant de près ou de loin .NET. Comme Mono ambitionne de reproduire totalement .NET (et pas seulement la partie standardisée), la contrefaçon semble inévitable. La situation est cependant suffisamment complexe pour qu'on se garde d'une conclusion hâtive sur les risques que

<sup>18</sup>La dernière version stable de Mono est la 1.1.13.8, sortie le 5 juin 2006

<sup>19</sup>la version 1.1.17, sortie fin août 2006, est considérée comme une beta bien avancée de Mono 1.2

court Mono, comme nous allons le voir dans la suite du texte.

#### 4.2.1 Red Hat et Mono

On sait que Red Hat, en tant qu'entreprise américaine, est particulièrement vulnérables aux risques induits par les brevets logiciels. Elle n'inclut donc pas dans sa distribution Fedora Core (ni bien sûr dans Red Hat Entreprise Linux) de logiciels susceptibles de violer des brevets, le cas le plus emblématique étant celui du format MP3 (cf au dessus).

Pendant longtemps, Fedora Core n'a pas contenu Mono. Les raisons de cette absence n'ont jamais été données explicitement, mais l'interprétation classique était que Mono risquait d'enfreindre des brevets de Microsoft sur .NET. Havoc Pennington, ancien président de la fondation GNOME et employé de Red Hat, indiquait dans son blog en mai 2005 [36] que Red Hat ne pouvait pas distribuer Mono et ne pouvait pas non plus détailler les raisons de cette impossibilité. Pourtant, en janvier 2006, Mono a été inclus dans *rawhide*, la version de développement de Fedora Core, puis dans la version 5 de la distribution [37].

Paradoxalement, le revirement de Red Hat n'est pas dû à une modification du statut légal de Mono, mais à la mise en place d'un mécanisme de protection que nous allons détailler.

#### 4.2.2 L'Open Invention Network

La décision de Red Hat a été expliquée, de façon officieuse, par un de ses employés, Greg DeKoenigsberg, dans son blog [38]. En résumé, cette décision s'appuie sur une politique de dissuasion qui n'est pas sans rappeler l'équilibre de la terreur de la dissuasion nucléaire.

Tout repose sur l'*Open Invention Network* (OIN [39, 40]). Cette entreprise a été créée en novembre 2005 sous l'impulsion de IBM, Novell, Philips, Red Hat et Sony. Le but de l'OIN est de réduire les risques que font peser les brevets logiciels sur Linux et plus généralement sur un certain nombre d'applications majeures liées à Linux.

Pour ce faire, l'OIN a délimité une zone "franche", c'est-à-dire une liste de logiciels protégés [41], qui va du noyau Linux à OpenOffice.org en passant par KDE, MySQL, etc. Pour devenir membre de l'OIN, une entreprise "paye" avec des licences : elle doit en effet s'engager à ne pas utiliser ses brevets (présents et à venir) contre les logiciels de la zone franche (noyau Linux et logiciels protégés). Les cinq membres fondateurs se sont donc engagés sur ce point.

Pour assurer que la zone franche reste sûre pour les développeurs, il faut inciter les entreprises à rejoindre l'OIN. L'entreprise achète donc des brevets logiciels dont elle offre une licence de type *royalty free* à ses membres. A ce jour, l'OIN possède 13 brevets [42], ainsi que quelques demandes, ce qui peut sembler assez faible en regard de l'effort à consentir pour rejoindre l'OIN. En fait, toute l'astuce réside dans le choix des brevets. Contrairement à ce qu'on pourrait croire, l'OIN n'a pas acquis des brevets couvrant directement Linux ou un des logiciels de la zone franche. Au contraire, les brevets ont été choisis de sorte à avoir une couverture assez large. Par exemple, certains brevets d'acquisition récente (mai 2006) portent plus ou moins directement sur les plateformes Java et .NET. C'est le cas du brevet de décembre 1998 numéro 5,848,274 (déposé en 1996), intitulé *Incremental byte code compilation system*, qui couvre le principe de l'ajout dynamique de code à un programme en cours d'exécution. Il s'applique vraisem-

blement au chargement dynamique de *byte code* en Java et en .NET. Un autre brevet concerne un système de programmation permettant de mélanger plusieurs langages. D'autres brevets, comme ceux de la défunte entreprise *Commerce One*, portent sur le commerce électronique et les services web, et semblent avoir une portée très large. Pour l'anecdote, c'est Novell, le principal sponsor de Mono, qui a acheté les brevets<sup>20</sup> de *Commerce One* en décembre 2004, pour la somme faramineuse de 15,5 millions de \$ US, avant de les offrir à l'OIN [43].

Bien entendu, cette incitation risque de rester encore trop faible pour que des entreprises comme Microsoft ou Sun rejoignent l'OIN. C'est ici qu'entre en jeu l'équilibre de la terreur. L'OIN a annoncé qu'elle ne chercherait pas à profiter de ses brevets de façon commerciale (elle ne vendra donc pas de licence), mais qu'elle ripostera contre toute attaque dirigée vers sa zone franche. En d'autres termes, si une entreprise tente de faire valoir un brevet contre Linux ou une des applications de la liste, l'OIN se réserve la possibilité de contre attaquer en utilisant ses propres brevets.

### 4.2.3 Et Mono ?

La bonne nouvelle pour Mono est que cette implémentation de .NET figure dans la liste des applications protégées, dans la zone franche de l'OIN. L'autre bonne nouvelle est que les brevets *Commerce One* ont, selon de nombreux experts, une portée assez large (cf le dossier des *Cover Pages* [44]). Pour les membres de l'OIN, notamment Red Hat, ces brevets semblent suffisamment dangereux pour être dissuasifs. L'OIN considère

---

<sup>20</sup>Notons que cette acquisition porte à la fois sur 3 brevets, mais aussi sur de nombreuses demandes dont deux ont déjà conduit à de nouveaux brevets.

donc qu'elle possède des armes pour défendre sa zone franche, ce qui a motivé l'inclusion de Mono dans Fedora Core.

Bien qu'il n'y ai pas eu beaucoup de communication à ce sujet, excepté une annonce de presse de l'OIN [45], l'acquisition de brevets portant sur les technologies de machines virtuelles et de développement multi-langage me semble aussi un pas important dans la défense de Mono (et plus généralement d'implémentations libres de .NET).

En fait, la création de l'OIN modifie l'évaluation des risques qu'on peut associer à Mono (et plus généralement à l'utilisation d'une implémentation libre de .NET). L'analyse des brevets de Microsoft pouvait en effet faire craindre le pire (en l'occurrence la mise place par Microsoft d'une licence RAND incompatible avec une implémentation open source des standards ECMA, ainsi que des licences prohibitives pour les brevets couvrant la partie non standardisée). Les brevets de l'OIN couvrent en partie .NET et en partie les activités de Microsoft dans le domaine des services web. Attaquer Mono pour son implémentation du cœur de .NET (la partie standardisée) apparaît donc comporter quelques dangers pour Microsoft. Les risques sont donc maintenant partagés, ce qui, si on est convaincu par la doctrine de l'équilibre de la terreur, devrait avoir pour conséquence de les réduire des deux côtés. En ce sens, bien que Mono soit toujours couvert par des brevets de Microsoft, il ne court pas plus de risques que les autres logiciels que Microsoft pourrait attaquer, comme par exemple OpenOffice.org. Pour évaluer les risques résiduels, il faut donc évaluer l'intérêt que pourrait avoir Microsoft à attaquer Mono.

#### 4.2.4 Que pense Microsoft de Mono ?

Bien que la protection fournie à Mono par l'OIN soit intéressante, elle ne fait que rendre plus coûteux un éventuel procès contre Novell. Microsoft possédant des réserves d'argent considérables (en mai 2006, l'entreprise disposait de 35 milliards de \$ US directement mobilisables), cet obstacle n'est pas nécessairement infranchissable. Il est donc important d'analyser l'attitude de Microsoft vis à vis de Mono pour avoir une vue complète de l'environnement dans lequel cette implémentation de .NET évolue.

Cette analyse est difficile car Microsoft ne semble pas avoir de politique arrêtée sur Mono. Officiellement, les seules relations entre Novell et Microsoft au sujet de .NET sont celles induites par la participation de Novell à l'ECMA. Novell est en effet membre associé de l'organisme et participe au comité technique 39 qui spécifie entre autres C# et la CLI. Novell possède cependant moins de poids que Microsoft (qui est membre ordinaire) car l'entreprise n'a pas le droit de vote en assemblée générale. Miguel De Icaza a récemment confirmé (et regretté) dans un entretien [46] accordé à Sam Ramji de Port 25<sup>21</sup> que les relations officielles entre Novell et Microsoft au sujet de .NET se limitaient à cette participation commune à l'ECMA.

De plus, les déclarations d'employés de Microsoft ne les engagent pas beaucoup. On perçoit une certaine sympathie, mais aussi des éléments négatifs. En juin 2004, John Montgomery, directeur du marketing chez Microsoft disait [47] :

*“Although the Mono Project promises to simplify the process of building cross-platform applications, Microsoft does not view*

<sup>21</sup>Port 25 est un site de blogs des membres de l'Open Source Software Lab de Microsoft.

*Mono as a competitive threat to its own development tools business [...] Mono has taken a small subset of .Net and cloned—and it's unclear how much they've cloned and how good it is<sup>22</sup>.”*

En septembre 2005, S. "Soma" Somasegar, vice président de la *Developer Division* de Microsoft indiquait [48] :

*“Whenever somebody does something on the .NET framework, I get excited. The first thing that comes to mind is, hey, they see value in the .NET framework and they want to do something with it, so I'm excited about that. These guys now have made good progress in terms of reverse-engineering the .NET framework. Good for them. To me, it means that the .NET framework is that much more interesting to people who have different opinions, different ways of looking at things<sup>23</sup>.”*

Le fait qu'une partie de .NET soit standardisée n'empêche pas M. Montgomery de parler de clonage. Cela n'empêche pas non plus

<sup>22</sup>“Bien que le projet Mono promette une simplification du processus de création d'applications multi-plateformes, Microsoft ne voit pas Mono comme une menace contre son offre en outils de développement [...] Mono a cloné une petite partie de .Net – l'étendue et la qualité du clonage restent à déterminer.”

<sup>23</sup>“Je suis excité à chaque fois que quelqu'un réalise quelque chose à partir de la plateforme .NET. La première chose qui me vient à l'esprit c'est “Ah, ils ont réalisé l'intérêt de la plateforme .NET et ils veulent faire quelque chose avec,” donc ça me stimule. Ces gars ont maintenant fait de grands progrès dans leur “reverse engineering” de la plateforme .NET. C'est bien pour eux. Pour moi, cela signifie que la plateforme .NET est d'autant plus intéressante pour des gens qui ont des opinions diverses, diverses façons de voir les choses.”

M. Somasegar de parler de *reverse engineering*, activité prohibée par de nombreuses licences logicielles et interdites dans certains pays. M. Somasegar est cependant assez positif sur l'expérience .NET elle-même. C'est aussi le cas de M. Montgomery qui disait en décembre 2002 [49] :

*"The fact that Ximian is doing this work is great. It's a validation of the work we've done, and it validates our standards activities. Also, it has caused a lot of eyeballs in the open source community to be directed to .Net, which we appreciate<sup>24</sup>."*

Cette opinion est partagée par le créateur de .NET, Anders Hejlsberg, qui disait de Mono en juin 2005 [50] :

*"Oh, I think it's great, I think it's proof that standardization works, that the work that we've done in ECMA to standardize CLI [Common Language Infrastructure] and C# actually has [results] and we have seen completely independent third-party implementations of the infrastructure. So I think it's a good thing. I think it's more good for us than bad for us<sup>25</sup>."*

On trouve souvent chez les employés de Microsoft une attitude ambiguë à l'égard de

<sup>24</sup> "C'est très bien que Ximian fasse ce travail. C'est une validation de notre travail et de nos activités de standardisation. En outre, une conséquence de ce travail est que de très nombreuses paires d'yeux de la communauté open source se sont tournées vers .NET, ce qui est appréciable pour nous."

<sup>25</sup> "Ok, je pense que c'est super, que c'est une preuve que la standardisation fonctionne, que le travail que nous avons fait au sein de l'ECMA pour standardiser la CLI et C# a vraiment des résultats et que nous voyons là une implémentation de l'infrastructure totalement indépendante et réalisée par des tiers. Donc je pense que c'est une bonne chose. Je pense que ça nous fait plus de bien que de mal."

Mono. Outre Montgomery et Somasegar, on peut citer Brian Goldfarb, chef produit de tout ce qui concerne le web qui disait en juin 2005 [51] :

*"We're excited about what they're doing, but we don't support the Mono project. They are completely on their own<sup>26</sup>."*

Quand Tim Anderson, qui l'interrogeait, lui demanda si Microsoft s'opposait au projet, il répondit :

*"No. We are totally indifferent towards it. I think it's great what they're doing, but it's neither here nor there<sup>27</sup>."*

Du côté positif, on note que Mono (de même que de nombreux autres outils libres, comme DotGnu, par exemple) est mentionné dans la liste des outils C# disponible sur le site du *Microsoft Developer Network* [52]. Outre le récent entretien accordé à Port 25 déjà mentionné [46], De Icaza avait été interrogé par le MSDN [53] en 2001, bien longtemps avant la sortie de Mono 1.0.

Du côté négatif, on peut noter De Icaza a tenté deux fois de proposer une session (*Birds of a feather*, BOF) à la *Professional Developers Conference* de Microsoft. Les deux tentatives se sont soldées par des échecs. Les propositions de session BOF sont normalement mises aux votes, mais Microsoft a spécifiquement supprimé la proposition portant sur Mono (au moins en 2005 [54, 55]). Pourtant, De Icaza a présenté officiellement Mono au symposium Lang.NET [56] organisé fin juillet 2006 par Erik Meijer de Microsoft, ce qui ne fait que confirmer

<sup>26</sup> "Nous sommes excité par ce qu'ils font, mais nous ne soutenons pas le projet Mono. Ils ne peuvent compter que sur eux-mêmes."

<sup>27</sup> "Non. Cela nous indiffère. Je pense qu'ils font quelque chose de super, mais rien n'est terminé."

à mes yeux la schizophrénie de l'entreprise (un phénomène assez courant dans les très grandes structures!).

#### 4.2.5 Synthèse

L'attitude de Microsoft envers Mono reste donc peu claire. On note pourtant une tendance qui s'articule autour de deux points de vue. Pour les scientifiques et les ingénieurs (comme Meijer et Hejlsberg), Mono est une réussite technique, à la fois pour Novell qui montre un savoir faire très important, mais aussi pour Microsoft qui a su spécifier et documenter sa plateforme de façon suffisamment complète et claire pour qu'on puisse l'implémenter de façon totalement indépendante. Comme l'écrit Stephen Walli, ancien de chez Microsoft, l'entreprise a en certain sens *besoin* de Mono car [57]

*“As long as there is only one commercial implementation, there is no standard, and the customers know it regardless of the number of trees killed to produce specifications<sup>28</sup>.”*

Le deuxième point de vue est plus celui des décideurs qui, tout en reconnaissant l'intérêt technique de Mono, nient son impact commercial possible, soit en le dénigrant (*“it's neither here nor there”, dixit Goldfarb*), soit en laissant planer des menaces peu claires concernant les brevets (*“we have a patent or one pending that's essential to implementing the specification”, dixit Herman*).

Il est donc assez probable, de mon point de vue, que Microsoft ne cherche pas dans les années à venir à empêcher l'implémentation de la partie standardisée de .NET. Novell sera peut être obligé de signer un ac-

---

<sup>28</sup> *“Tant qu'il n'existe qu'une implémentation commerciale, il n'y a pas de standard, et les consommateurs le savent, quelle que soit la quantité d'arbres abattus pour imprimer les spécifications.”*

cord de licence, mais il n'y a pas de raison de croire que Microsoft cherche à mettre en péril son travail de standardisation en ne proposant pas une licence *Royalty Free*. Cette licence sera sûrement compatible avec une implémentation libre (mais très certainement pas avec une licence GPL [18]).

Par contre, il est possible que Microsoft tente de faire valoir ses brevets sur certaines parties de l'API .NET, surtout si Mono devient trop concurrentiel. On sait, par exemple, que Microsoft verrait d'un très mauvais œil une implémentation open source de deux des futures briques de base de Windows Vista, *Windows Communication Foundation* (ex Indigo) et *Windows Presentation Foundation* (ex Avalon), cf par exemple [58, 59, 60]. Indigo consiste essentiellement en un ensemble d'outils permettant la mise en place de services web [61]. Une partie d'Indigo est donc standardisée par le W3C et OASIS. Cette partie peut être implémentée librement, ce qui permet de se connecter à un service web fourni par un serveur Indigo. C'est justement le côté serveur que Microsoft entend protéger, notamment grâce à ses brevets. Avalon [62], de son côté, peut être vu comme un moteur et une api de rendu graphique très sophistiqués, dans l'esprit de Cairo [63], associés à un dialecte XML permettant de décrire des interfaces graphiques.

Cependant, ceci ne semble pas faire courir un grand risque à Mono. Pour l'instant, ces deux briques ne sont pas encore officiellement disponibles. Le projet Mono s'intéresse à Indigo et Novell a démarré un projet plus ou moins compatible, appelé Amber [64]. Le projet ne semble cependant pas de faire un clone d'Indigo mais plutôt de supporter les mêmes standards. Miguel De Icaza n'est pas un grand admirateur d'Avalon (cf [65]) et aucune tentative de clonage du moteur de rendu n'est pour l'instant envisagée [66].

## 5 Java

### 5.1 Rappels sur la plateforme

Contrairement à .NET, Java n'est pas une plateforme standardisée par un organisme indépendant. La spécification et l'évolution de la plateforme sont confiées au *Java Community Process* (JCP, [67]). Ce programme permet une collaboration entre Sun et ses partenaires, en majorité des grandes entreprises comme IBM ou SAP, mais aussi la fondation Apache, le groupe JBoss et le professeur Doug Lea de l'Université de New York Oswego (cf [68]). Le JCP produit des *Java Specification Requests* (JSR) qui décrivent des évolutions de la plateforme, c'est-à-dire du langage Java, de la machine virtuelle et des API. Le JCP s'intéresse tout particulièrement aux trois profils que compte la plateforme, à savoir l'édition embarquée dite *Micro Edition* (pour les téléphones portables, par exemple), l'édition classique dite *Standard Edition* et l'édition entreprise (*Enterprise Edition*). Un profil est défini par la liste des API supportées.

Je ne reviendrais pas en détail ici sur le fonctionnement du JCP car il est déjà présenté de façon assez complète dans mon précédent article [1] et que la situation n'a pas évolué de façon sensible depuis.

### 5.2 Une implémentation libre possible si elle est conforme

L'avantage de Java sur .NET est que le créateur du premier s'est engagé depuis longtemps à autoriser des implémentations libres (et prévoit d'aller beaucoup plus loin, comme nous allons le voir). Comme je l'indiquais dans [1], la participation au JCP n'est possible qu'en signant le JSPA (Java Specification Participation Agreement, [69]). Celui-ci impose (section 5, en particulier

le point B) à tout JSR d'accorder une licence d'implémentation et de distribution sans redevance (*Royalty Free*). La formulation du JSPA dans sa version 2<sup>29</sup> a été pensée pour satisfaire les demandes de la fondation Apache [70, 71], et notamment pour assurer la possibilité de réaliser une implémentation libre de chaque JSR.

Il y a cependant une contrainte de taille. Une implémentation (libre ou non) d'un JSR doit obligatoirement être conforme, c'est-à-dire

1. qu'elle doit implémenter complètement la spécification ;
2. qu'elle ne doit pas ajouter, modifier ou supprimer quelque chose dans les packages définis par la spécification ;
3. qu'elle doit passer avec succès les tests de compatibilité (le *Technology Compatibility Kit*, un ensemble de tests de conformité associé à chaque JSR).

Bien que ceci ne soit pas *stricto sensu* incompatible avec une implémentation libre, cela rend celle-ci plus difficile, comme l'a bien analysé Richard Stallman dans [8]. Pour résumer (cf aussi [1]), la diffusion d'une version incomplète d'une implémentation d'un JSR, voire d'une implémentation qui n'a pas effectivement passé le TCK (par exemple parce qu'on n'en possède pas une licence), est potentiellement illégale (car elle ne respecte pas les conditions d'obtention d'une licence d'implémentation ou de distribution). De ce fait, le développement ouvert des logiciels libres est officiellement interdit, sauf si le "public" ne peut accéder qu'à des versions passant le TCK (par exemple grâce à un mécanisme de test automatique qui ne publie une image du projet que si elle est conforme). Notons que le problème légal est ici double :

---

<sup>29</sup>la version actuelle est la 2.0.1

1. le contenu de la spécification d'un JSR est soumis aux règles classiques du *copyright* et du droit des marques (l'utilisation du mot Java n'est pas libre, par exemple) ;
2. toute implémentation d'un JSR risque de violer des brevets sur certains de ces aspects pour lesquels il faut donc obtenir une licence (contenue dans celle qu'on accepte pour obtenir la spécification).

Il faut aussi noter que même si une implémentation ne prétend pas être une réalisation conforme d'un JSR, elle peut ne pas respecter le copyright et/ou le droit des marques le concernant, ou encore contrefaire un brevet le protégeant.

En pratique cependant, Sun et ses partenaires n'ont jamais posé de problèmes aux projets libres, et certaines implémentations de JSR ont acquis le statut officiel d'implémentations conformes. Les cas les plus médiatisés sont certainement ceux de JBoss [72] et de JOnAS [73] pour leur implémentations libres<sup>30</sup> de la version entreprise de la plateforme Java (J2EE 1.4), mais il ne faut pas oublier les contributions majeures de la fondation Apache [74], comme par exemple Tomcat [75], implémentation de références des Servlet et des Java Server Pages, Geronimo [76] implémentation conforme de J2EE 1.4, les très nombreux éléments inclus dans l'implémentation de référence de J2SE par Sun (Xerces pour l'analyse de fichiers XML, Xalan comme implémentation du langage de transformation XSLT, BCEL pour la manipulation du byte code des fichiers .class, Regexp comme moteur d'analyse d'expressions régulières), etc.

---

<sup>30</sup>Ces logiciels complètent le profil standard J2SE de Sun par une implémentation libre des API additionnelles permettant d'atteindre le profil entreprise.

Cependant, tous ces logiciels libres dépendent d'une implémentation du profil standard de la plateforme Java (J2SE) pour lequel il n'existe pas encore<sup>31</sup> d'implémentation libre complète. La licence d'implémentation et de distribution du J2SE étant la même (depuis la version 1.4) que celle du J2EE (aussi depuis la version 1.4), il est parfaitement possible de réaliser légalement une implémentation libre du premier profil comme cela a déjà été fait pour le second. Le travail considérable que cela représente (en raison principalement de la taille de l'api du J2SE) explique que le travail reste pour l'instant inachevé. Ceci devrait changer dans les mois qui viennent, tant en raison des progrès réalisés par les implémentations libres existantes que de la sortie prochaine de la version de Sun en open source.

### 5.3 Les implémentations libres de Java

Quand on parle d'implémentation libre de Java, on fait référence au profil standard, au J2SE qui a déjà connu 6 versions depuis la 1.0 jusqu'à la 1.5 actuellement stable (la version 1.6 est déjà disponible en version beta). Dans ce profil, la plateforme comprend une machine virtuelle et une énorme api. On lui ajoute en général la partie qui permet le développement Java, à savoir essentiellement un compilateur (ainsi que d'autres éléments moins cruciaux, comme des fichiers *headers* pour l'interfaçage avec le C, par exemple). Le J2SE associé aux éléments nécessaires au développement forme le JDK (*Java Development Kit*).

---

<sup>31</sup>début septembre 2006.

### 5.3.1 Classpath

Les implémentations les plus avancées du J2SE sont basées sur le projet GNU Classpath [77] qui vise à implémenter l'intégralité de l'api du J2SE. Le but de la version 1.0 de Classpath est d'être compatible avec la version 1.1 du J2SE. La version actuelle (0.92) implémente 99.78% de l'api 1.1. La couverture de l'api 1.4 (la première à être officiellement implémentable librement) est de 99.85%.

Au niveau des machines virtuelles, la concurrence est importante. Parmi celles donc le développement est très actif et suit de près les *releases* de Classpath, on peut citer Kaffe [78], CACAO [79], SableVM [80], Jikes RVM [81], IKVM.NET [82] et JamVM [83]. Toutes ces solutions fonctionnent relativement bien, incluent généralement une compilation au vol (*just in time*) et implémentent de façon relativement complète les spécifications de la machine virtuelle. On peut considérer que Kaffe et IKVM.NET sont les solutions développées les plus activement.

La compilation de sources Java vers le bytecode de la machine virtuelle a longtemps été assurée par le compilateur open source d'IBM, Jikes [84]. Celui-ci n'a cependant pas évolué depuis octobre 2004 et ne supporte pas les ajouts au langage de la version 1.5. On lui préfère donc le compilateur inclus dans l'environnement de développement Eclipse [85].

Le projet GCJ (intégré à GCC [86]) propose une solution très originale dans le monde Java, qui permet de compiler directement du Java vers du code natif, tout en conservant tous les aspects dynamiques de la plateforme, en particulier du chargement de bytecode à la volée.

Les implémentations libres de Java basées

sur Classpath sont très avancées. Comme Mono, elles offrent une plateforme de développement complète, en particulier grâce à SWT [87], une bibliothèque graphique libre concurrente du Swing intégré à J2SE. De nombreuses applications complexes tournent très bien après compilation par GCJ. C'est le cas par exemple d'Eclipse, du client BitTorrent Azureus [88], du serveur JOnAS mentionné plus haut, etc. Les progrès récents dans l'implémentation libre de Swing ont permis de faire fonctionner des applications développées pour cette api, comme par exemple jEdit [89] et JFreeChart [90].

Notons cependant que, *stricto sensu*, aucune de ces implémentations n'est légale puisque Classpath ne passe pas les tests de conformité.

### 5.3.2 Harmony

En 2005, Geir Magnusson Jr. lance le projet Harmony, incubé au sein de la fondation Apache [91]. Le but du projet est la réalisation d'une implémentation sous licence Apache du J2SE dans sa version 1.5, ce qui inclut donc une machine virtuelle, un compilateur et une implémentation de l'api. A son lancement, le projet ne semblait pas trop ambitieux, car une collaboration avec Classpath était envisagée. Cependant, l'incompatibilité entre la GPL (dans sa version 2) et la licence Apache, ainsi qu'une certaine absence de souplesse de la part de la fondation [92], ont rendu toute collaboration impossible, ce qui fait d'Harmony un projet totalement indépendant de Classpath. Les tensions qui sont apparus pendant les négociations à ce sujet ont finalement provoqué le départ de la plupart des projets libres qui pensaient initialement collaborer avec Harmony [93]. Cependant, Harmony

est très fortement soutenu par de grosses entreprises, en particulier IBM et Intel. Il a reçu des dons de code très conséquents, en particulier une implémentation d'AWT, Java2D et Swing ainsi qu'une machine virtuelle et des outils associés en provenance d'Intel (IBM a fait don de classes fondamentales). De ce fait, et malgré la mésentente avec les projets plus anciens, Harmony progresse relativement vite et est déjà capable de faire tourner quelques applications comme jEdit. La couverture de l'api 1.5 (89.44%), reste inférieure à celle de Classpath (95.13%), mais les progrès réalisés en à peine plus d'un an sont considérables (techniquement, l'implémentation proposée par Harmony n'est toujours pas légale, la conformité n'étant pas assurée).

### 5.3.3 Et l'implémentation de Sun ?

Malgré ses efforts pour autoriser l'implémentation d'une version libre de Java, Sun a longtemps refusé de rendre libre sa propre implémentation de référence. En réponse à des pressions d'IBM à ce sujet [94] (quelques jours après une demande similaire d'Eric Raymond [95]), Sun répondait vouloir conserver un contrôle serré de Java pour éviter tout problème de compatibilité arguant, contre toute vraisemblance, qu'une licence open source favorisait les *forks* [96].

Cependant, Sun a progressivement changé son discours. En mars 2005, Sun a ainsi lancé de nouvelles licences pour le code source de son implémentation de J2SE et pour la version binaire de celle-ci [97]. Considérées comme très complexes (cf [98, 99] par exemple), elles n'en constituaient pas moins un pas en avant vers un peu plus d'ouverture, confirmé par une amélioration de la licence de distribution (dans le sens d'un surcroît d'ouverture) en mai 2006 [100].

L'étape suivante a été l'annonce prononcée en mai 2006 par le PDG de Sun, Jonathan Schwartz, à l'occasion de la conférence JavaOne [101] :

*“The question is not whether we will open-source Java, the question is how.”<sup>32</sup>*

L'engagement a été confirmé en août 2006 lors de la conférence LinuxWorld. Sun prévoit de mettre ses implémentations de référence des profils J2SE et J2ME sous une licence libre (approuvée par l'OSI) dans les mois qui viennent [102]. L'entreprise prévoit d'abord de placer son compilateur et sa machine virtuelle sous une licence libre en octobre 2006, puis la version J2ME complète pour la fin 2006. Le reste du profil J2SE sera libéré début 2007, à l'exception des éléments que Sun ne pourra pas mettre sous licence libre pour des raisons de droits partagés sur le code [103]. Pour mieux communiquer sur cet événement majeur du monde Java, Sun a créé à l'occasion de cette annonce un site dédié à la libération de la plateforme, sur lequel on trouve une agrégation de blogs, une revue de presse, un forum, etc. [104].

L'annonce de Sun a provoqué et provoque encore beaucoup de réactions et de débats, dont le site créé par l'entreprise donne un écho très fidèle. L'essentiel des débats porte sur la licence que Sun va choisir pour son implémentation. Son caractère libre est acquis, mais rien d'autre n'est arrêté officiellement. Il est possible, par exemple, que la licence choisie ne soit pas compatible avec la GPL, ce qui empêcherait toute collaboration avec le projet Classpath. La licence pourrait aussi être incompatible avec la licence Apache, bloquant cette fois-ci toute collaboration avec Harmony. L'idéal pour le libre

<sup>32</sup>*“La question n'est pas de savoir si nous allons rendre Java open-source, mais plutôt comment nous allons le faire.”*

serait bien entendu une licence compatible à la fois avec la GPL et la licence Apache, ce qu'appellent de leurs vœux les développeurs Classpath comme Tom Tromeu [105].

## 5.4 Java et les brevets

### 5.4.1 Brevets détenus par Sun

Tout comme .NET, Java est couvert par des brevets logiciels. James Gosling, inventeur de Java, est par exemple co-auteur de très nombreux brevets portant sur la plateforme et possédés par Sun. Neuf d'entre eux portent sur la vérification de byte code, qui est au cœur du modèle de sécurité de Java, d'autres concernent les servlets, sur l'interface graphique, sur l'aspect réparti, le chargement dynamique de byte code, etc. Plus généralement, on compte 89 brevets possédés par Sun et mentionnant Java dans leur résumé. Bien que Sun n'ait jamais voulu donner une liste de ses brevets s'appliquant à Java (cf [106], page 5), il est clair que ces derniers sont nombreux et que leur couverture est très large : on voit mal comment une implémentation du J2SE pourrait ne pas enfreindre ces brevets.

Cependant, contrairement au cas de .NET, l'utilisation de ces brevets est officiellement permise pour réaliser une implémentation du J2SE. Pour télécharger la spécification de ce profil de Java, il faut en effet accepter la licence correspondante [107] qui donne explicitement un droit d'implémentation (sous réserve de conformité). Cette licence est compatible avec une implémentation libre. Rappelons qu'au contraire, Microsoft n'a toujours pas publié de document officiel précisant la licence qu'il compte accorder pour l'implémentation des standards ECMA.

### 5.4.2 Autres brevets

Malheureusement, la grande popularité de la plateforme Java a déchaîné les "inventeurs". On compte en effet 552 brevets possédés par d'autres entreprises que Sun qui mentionnent Java dans leur résumé. Ces brevets sont cependant construits autour de la technologie Java et font donc référence aux brevets détenus par Sun. En ce sens, enfreindre un des brevets implique généralement d'enfreindre ceux de Sun. De ce fait, Sun lui-même est relativement protégé des menaces induites par les brevets des autres inventeurs. Le cas d'une implémentation libre indépendante est moins clair car on rencontre ici les brevets généraux que je mentionnais en début d'article. Il faudrait analyser les brevets un à un pour savoir s'ils couvrent des algorithmes mis en œuvre dans une implémentation libre.

Il existe cependant des menaces plus graves car tous les brevets qui s'appliquent à Java ne sont pas postérieurs à ceux de Sun. Un exemple édifiant est celui des brevets "Kodak". En 2002 Kodak a attaqué Sun au sujet de trois brevets, de formulation assez obscure, qui couvrent vraisemblablement l'utilisation d'un bus logiciel pour faire collaborer des objets (et même plus généralement la notion de délégation d'une tâche d'un objet à un autre) et qui s'appliquent de ce fait à Java [108]. Sun a été reconnu coupable de contrefaçon par un jury populaire<sup>33</sup> [109], puis est parvenu à un accord de licence avec Kodak portant sur un montant de 92 millions de \$ US [110, 111]. Les termes exacts de l'accord ne sont pas

---

<sup>33</sup>A la lecture des brevets, je me permets de me poser des questions sur la pertinence d'un tel jugement, non pas sur le caractère avéré (ou non) de la contrefaçon, mais plutôt au sujet de la compétence technique du jury et donc des motivations du jugement.

connus, mais Sun a indiqué que ses clients étaient protégés [112]. L'entreprise a aussi mentionné dans son communiqué de presse les projets libres auxquels elle participait à l'époque (comme OpenOffice.org), en indiquant clairement que les produits basés sur Java dans le cadre d'un accord de licence avec Sun étaient aussi couverts par l'accord avec Kodak. Il est donc probable que le futur JDK open source sera protégé par cet accord. Il n'est pas clair que cela soit aussi le cas des implémentations libres indépendantes. Notons au passage que selon divers experts (cf [110] par exemple), les brevets Kodak s'appliquent à .NET et que Microsoft possède un accord de licence à leurs sujets. Il n'y pas de raison particulière pour que Mono soit protégé par cet accord, ce qui le rend au moins aussi vulnérable de Classpath et Harmony.

## 6 Conclusion

Comme dans mon précédent article sur le même sujet [1], je me garderai bien d'émettre une conclusion définitive sur les plateformes Java et .NET. Je me contenterai de proposer un point de vue personnel d'amateur.

En un certain sens, les (presque) deux années qui nous séparent de ma première analyse ont été très bénéfiques pour le libre. Du point de vue technique, les implémentations *open source* des deux plateformes ont énormément progressé au point de proposer des alternatives viables aux implémentations de référence.

Du point de vue légal, la situation est devenue un peu plus claire. Concernant .NET, on sait maintenant avec certitude que Microsoft possède des brevets couvrant la plateforme (à la fois la partie standardisée et le reste). La création de l'Open Invention Network semble avoir réduit le risque de

voir Microsoft attaquer Novell a sujet de Mono, son implémentation libre de .NET. Des considérations stratégiques renforcent l'impression que Mono ne risque pas grand chose, au moins pour son implémentation du standard. En outre, les applications les plus intéressantes construites pour Mono (Beagle, F-Spot, etc.) n'utilisent pas ou peu les parties les moins sûres juridiquement (celles qui implémentent des api Microsoft non standardisées comme Windows.Forms ou ASP.NET). En fait, la situation semble s'être suffisamment clarifiée pour que Red Hat se décide à inclure Mono dans Fedora Core (depuis la version 5) et que le projet Gnome lui offre une place officielle (un nouveau module officiel de Gnome peut maintenant dépendre de `gtk#/mono`, cf [113]). Il faudrait être naïf pour croire que tous les problèmes de Mono sont réglés, mais les risques qu'on prend en développant avec cette plateforme sont moins importants qu'ils ne semblaient l'être fin 2004.

Pour Java, la situation est encore plus claire puisque Sun va libérer son implémentation de référence dans les mois qui viennent, ce qui va réduire de façon drastique les risques inhérents à cette plateforme. En outre, les implémentations libres concurrentes ne risquent pas grand chose de Sun, même si elles ne respectent pas encore la lettre de l'accord de licence associé aux spécifications du J2SE, car elles en suivent l'esprit : préserver une compatibilité complète avec l'implémentation de référence.

Cependant, les menaces que font peser les brevets logiciels sur le libre et, plus généralement, sur la création logicielle, sont loin d'être réglées, quelle que soit la bonne volonté d'acteurs comme Sun ou la protection offerte par des structures comme l'OIN. J'en veux pour preuve le procès intenté par FireStar Software contre Red Hat sur une

allégation de contrefaçon [114] que réaliserait le projet Hibernate [115]. FireStar détient un brevet portant sur le *mapping relationnel objet*, c'est-à-dire sur les algorithmes qui traduisent une classe en un schéma de base de données, pour stocker des objets dans un SGBD. Le projet Hibernate propose un outil de *mapping* à la fois pour Java et pour .NET, et pourrait donc enfreindre le brevet (comme tout outil de ce genre, tant les revendications sont larges). On en sait pas encore quelle sera la stratégie de Red Hat concernant ce procès, mais il marque la concrétisation d'une des plus grandes peurs de notre communauté : il s'agit en effet du premier procès qui oppose à un projet open source une accusation de contrefaçon [116].

L'écriture d'un logiciel libre est donc une activité risquée dans un monde qui autorise la privatisation du savoir, sous le prétexte fallacieux de favoriser l'innovation. Pour ne pas céder à un sentiment d'impuissance ou de panique, il faut comprendre la hiérarchie des risques, et savoir ainsi qu'on en prend un peu plus en choisissant de développer avec Mono qu'en s'appuyant sur le JDK de Sun. De même que le choix d'une licence libre n'est pas sans conséquence sur le futur d'un projet (appropriation commerciale, support institutionnel, financement, etc.) et doit donc être fait en connaissance de cause, celui d'un environnement de développement ne devrait pas seulement reposer sur des arguments techniques. J'espère que cet article aiderait chacun à prendre ses responsabilités et à arbitrer entre la technique (rapidité de développement, performances du logiciel final, etc.) et le juridique.

## Références

- [1] Fabrice Rossi. Java, .NET et les logiciels libres. *GNU/Linux Magazine France*, (67) :8–17, Décembre 2004. <http://apiacoa.org/publications/2004/lm-java-dotnet-free.pdf>.
- [2] Site de l'Open Source Initiative. <http://www.opensource.org/>.
- [3] Site de ECMA International. <http://www.ecma-international.org/>.
- [4] Site de l'International Organization for Standardization (iso). <http://www.iso.ch/>.
- [5] ECMA international. *C# Language Specification*, fourth edition, June 2006. Standard ECMA-334, disponible à l'URL <http://www.ecma-international.org/publications/standards/Ecma-334.htm>.
- [6] ECMA international. *Common Language Infrastructure (CLI)*, fourth edition, June 2006. Standard ECMA-335, disponible à l'URL <http://www.ecma-international.org/publications/standards/Ecma-335.htm>.
- [7] Code of conduct in patent matters. <http://www.ecma-international.org/memento/codeofconduct.htm>. La politique de l'ECMA en matière de brevets.
- [8] Richard Stallman. Free but shackled - the Java trap. <http://www.gnu.org/philosophy/java-trap.html>, April 2004.
- [9] Thomson. Patent portfolio. <http://www.mp3licensing.com/patents/index.html>.
- [10] Gabriel Bouvigne. Patents and mp3. <http://www.mp3-tech.org/patents.html>.
- [11] Article mp3 sur la wikipédia. <http://en.wikipedia.org/wiki/MP3>.
- [12] Mp3 support in fedora core. <http://fedoraproject.org/wiki/Multimedia/MP3> et [http://fedoraproject.org/wiki/ForbiddenItems#MP3\\_Support](http://fedoraproject.org/wiki/ForbiddenItems#MP3_Support). Fedora Wiki.
- [13] Free Software Foundation. Some confusing or loaded words and phrases that are worth avoiding. <http://www.gnu.org/philosophy/words-to-avoid.html>.
- [14] Thomson. Royalty rates. <http://www.mp3licensing.com/royalty/>.
- [15] Site du projet lame. <http://lame.sourceforge.net/>.
- [16] Seth Nickell. Why mono is currently an unacceptable risk. <http://www.gnome.org/~seth/blog/mono>, May 2004.
- [17] ASF position regarding Sender ID. <http://apache.org/foundation/docs/sender-id-position.html>, September 2004.
- [18] David Berlind. Will C# benefit Microsoft, or the industry? <http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2887217,00.%html>, October 2002. ZDNet.
- [19] Grégoire Lejeune. Introduction au langage c#. *GNU/Linux Magazine Franc*, (67), Décembre 2004.
- [20] Steve ballmer : Kein tänzchen an der leine. <http://www.heise.de/newsticker/meldung/25564>, Mars 2002. Heise Online.

- [21] 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.
- [22] Site du projet Mono. <http://www.mono-project.com/>.
- [23] Martin LaMonica. More than an open-source curiosity. [http://news.com.com/2008-7344\\_3-5271084.html](http://news.com.com/2008-7344_3-5271084.html), July 2004. CNET News.com.
- [24] John Sequeira. Mono developer meeting. [http://www.oreillynet.com/databases/blog/2004/03/mono\\_developer\\_meeting%.html](http://www.oreillynet.com/databases/blog/2004/03/mono_developer_meeting%.html), March 2004.
- [25] Adam W. Smith et al. Application program interface for network software platform. Patent Application 10/087,027, United States Patent and Trademark Office, February 2002. Numéro de publication 20030028685. Disponible sur le site <http://portal.uspto.gov/external/portal/pair>.
- [26] Andrew Orlowski. Ms patents .Everything. [http://www.theregister.co.uk/2003/02/11/ms\\_patents\\_everything/](http://www.theregister.co.uk/2003/02/11/ms_patents_everything/), February 2003. The Register.
- [27] Lisa M. Bowman. .Net patent could stifle standards effort. [http://news.com.com/2100-1001\\_3-984052.html](http://news.com.com/2100-1001_3-984052.html), February 2003. CNET News.com.
- [28] Andrew F. Roberts et al. Method and apparatus for providing web based services using an xml runtime model to store state session data. Patent 6,792,605, United States Patent and Trademark Office, September 2004.
- [29] David Yach. Virtual machine web browser. Patent Application 09/728,543, United States Patent and Trademark Office, August 2002. Numéro de publication 20020112078.
- [30] Richard Louis Firth and David Treadwell. System having a library of protocol independent reentrant network interface functions for providing common calling interface for communication and application protocols. Patent 5,987,517, United States Patent and Trademark Office, November 1999.
- [31] Dee-Ann LeBlanc. New mono-based applications for gnome in fedora core 5. <http://www.linuxplanet.com/linuxplanet/reviews/6232/1/>, June 2006. Linux Planet.
- [32] Site du projet DotGNU. <http://www.dotgnu.org/>.
- [33] Site de Beagle. [http://beagle-project.org/Main\\_Page](http://beagle-project.org/Main_Page).
- [34] Site de F-Spot. [http://www.f-spot.org/Main\\_Page](http://www.f-spot.org/Main_Page).
- [35] Site de Tomboy. <http://www.beatniksoftware.com/tomboy/>.
- [36] Havoc Pennington. Yay, language debates. <http://log.ometer.com/2005-05.html#10>, May 2005.
- [37] Stephen Shankland. Red hat : Mono in fedora, not enterprise linux. [http://news.com.com/2061-10795\\_3-6025387.html](http://news.com.com/2061-10795_3-6025387.html), January 2006. CNET News.com.
- [38] Greg DeKoenigsberg. Fedora and Mono and OIN – clarifications. <http://gregdek.livejournal.com/4008.html>, March 2006.
- [39] Site de l'Open Invention Network. <http://www.openinventionnetwork.com/>.

- [40] Mark H. Webbink. The Open Invention Network. *Linux Magazine*, page 18, April 2006. Consultable en ligne après enregistrement gratuit sur <http://www.linux-mag.com/content/view/2541/>.
- [41] Open Invention Network. Complete list of linux environment components. [http://www.openinventionnetwork.com/pat\\_linuxdefpop.html](http://www.openinventionnetwork.com/pat_linuxdefpop.html), 2006.
- [42] Open Invention Network. Open Invention Network's currently owned patents. [http://www.openinventionnetwork.com/pat\\_owned.php](http://www.openinventionnetwork.com/pat_owned.php).
- [43] John Markoff. Novell discloses it bought e-commerce patents. *The New York Times*, May 2005. <http://www.ihf.com/articles/2005/05/02/business/novell.php>.
- [44] Robin Cover. Commercenet proposes collecting contributions to purchase key web services patents. <http://xml.coverpages.org/ni2004-11-24-a.html>, November 2004. Cover Pages.
- [45] Open invention network acquires new patents to protect linux. [http://www.openinventionnetwork.com/press\\_release.php](http://www.openinventionnetwork.com/press_release.php), May 2006.
- [46] Sam Ramji. Talking mono with miguel de icaza. [http://port25.technet.com/archive/2006/08/11/Let\\_2700\\_s-talk-Mono\\_3A00\\_%--Sam-interviews-Miguel-de-Icaza.aspx](http://port25.technet.com/archive/2006/08/11/Let_2700_s-talk-Mono_3A00_%--Sam-interviews-Miguel-de-Icaza.aspx), August 2006. Port 25.
- [47] Martin LaMonica. Novell ships cross-platform mono tool. [http://news.com.com/Novell+ships+cross-platform+Mono+tool/2100-1012\\_3-5%253448.html](http://news.com.com/Novell+ships+cross-platform+Mono+tool/2100-1012_3-5%253448.html), June 2004. CNET News.com.
- [48] Todd Bishop. The mono 'experiment'. <http://blog.seattlepi.nwsourc.com/microsoft/archives/005691.html>, September 2005. Todd Bishop's Microsoft Blog.
- [49] John McGrath. Mono & .net : The odd couple. <http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2907356,00.%html>, December 2002. ZDNet.
- [50] Paul Krill. Microsoft's hejlsberg touts .net, c-omega technologies. [http://www.infoworld.com/article/05/06/10/HNhejlsberg\\_1.html](http://www.infoworld.com/article/05/06/10/HNhejlsberg_1.html), June 2005. InfoWorld.
- [51] Tim Anderson. Miguel de icaza on mono. <http://www.itwriting.com/monointerview.php>, 2005 July.
- [52] C# programming tools. <http://msdn.microsoft.com/vcsharp/programming/tools/>. MSDN.
- [53] Dare Obasanjo. Using the ecma standards : An interview with miguel de icaza. <http://msdn.microsoft.com/netframework/programming/clr/default.aspx?pul%1=/library/en-us/dndotnet/html/deicazainterview.asp>, December 2001. MSDN.
- [54] Todd Bishop. Rival shines in shadows at microsoft's l.a. event. *Seattle Post-Intelligencer*, September 2005. [http://seattlepi.nwsourc.com/business/240989\\_pdcmono16.html](http://seattlepi.nwsourc.com/business/240989_pdcmono16.html).
- [55] Miguel de Icaza. Mono meeting at the microsoft pdc. <http://tirania.org/blog/archive/2005/Sep-06.html>, September 2005.

- [56] Lang.net symposium. <http://www.langnetsymposium.com/>, July–August 2006.
- [57] Stephen Walli. Mono, microsoft, and mischief. [http://stephesblog.blogs.com/my\\_weblog/2005/08/mono\\_microsoft\\_.html](http://stephesblog.blogs.com/my_weblog/2005/08/mono_microsoft_.html), August 2005.
- [58] Gavin Clarke. Indigo not so open as .net framework? [http://www.regdeveloper.co.uk/2005/05/26/ndigo\\_mono\\_no\\_no/](http://www.regdeveloper.co.uk/2005/05/26/ndigo_mono_no_no/), May 2005. Reg Developer.
- [59] Gavin Clarke. Pay to play with microsoft’s indigo. [http://www.theregister.co.uk/2005/06/16/\\_microsoft\\_indigo/](http://www.theregister.co.uk/2005/06/16/_microsoft_indigo/), June 2005. The Register.
- [60] Steven J. Vaughan-Nichols. Microsoft puts roadblock in front of open-sourcing avalon and indigo. *eWeek*, June 2005. <http://www.eweek.com/article2/0,1759,1829797,00.asp>.
- [61] Windows communication foundation. <http://msdn.microsoft.com/winfx/technologies/communication/default.aspx%>. Microsoft .NET Framework 3.0.
- [62] Windows presentation foundation. <http://msdn.microsoft.com/winfx/reference/presentation/default.aspx>. Microsoft .NET Framework 3.0.
- [63] Site de cairo. <http://cairographics.org/>.
- [64] Projet amber sur novell forge. <http://forge.novell.com/modules/xfmod/project/?amber>.
- [65] Miguel de Icaza. A j2ee moment of zen. <http://tirania.org/blog/archive/2006/Aug-02.html>, August 2006.
- [66] Mono hacking roadmap. [http://www.mono-project.com/Mono\\_Hacking\\_Roadmap#Avalon\\_plans](http://www.mono-project.com/Mono_Hacking_Roadmap#Avalon_plans), September 2005.
- [67] Site du Java Community Process (JCP). <http://www.jcp.org/>.
- [68] Executive Committee Info. <http://jcp.org/en/participation/committee>. Composition de l’Executive Committee du JCP.
- [69] Java Specification Participation Agreement. <http://www.jcp.org/aboutJava/communityprocess/JSPA2.pdf>, Janvier 2005. Version actuellement en vigueur : 2.0.1.
- [70] JSPA news & status. <http://jakarta.apache.org/site/jspa-position.html>, 2002. La position de la fondation Apache pour la révision du JSPA.
- [71] 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.
- [72] JBoss becomes first open source application server to achieve J2EE 1.4 compatibility certification. <http://www.jboss.com/pdf/press/j2eecertfinal.pdf>, July 2004. Annonce de presse de JBoss Inc. pour la certification de JBoss 4.0 en tant que serveur J2EE.
- [73] JOnAS becomes the first non commercial open-source application server to complete J2EE 1.4(tm) compatibility certification. <http://www.objectweb.org/phorum/read.php?f=25&i=88&t=88>, February 2005. Annonce de presse du consortium ObjectWeb pour la certification de JOnAS 4 en tant que serveur J2EE.

- [74] Apache and the jcp. <http://www.apache.org/jcp/>. Description de la participation de la fondation Apache au JCP.
- [75] Site de Tomcat. <http://tomcat.apache.org/>.
- [76] Site de Geronimo. <http://geronimo.apache.org/>.
- [77] Site de Classpath. <http://www.gnu.org/software/classpath/>.
- [78] Site de Kaffe. <http://www.kaffe.org/>.
- [79] Site de CACAO. <http://www.cacaojvm.org/>.
- [80] Site de SableVM. <http://sablevm.org/>.
- [81] Site de Jikes RVM. <http://jikesrvm.sourceforge.net/>.
- [82] Site de IKVM.NET. <http://www.ikvm.net/>.
- [83] Site de JamVM. <http://jamvm.sourceforge.net/>.
- [84] Site de Jikes. <http://jikes.sourceforge.net/>.
- [85] Site d'eclipse. <http://www.eclipse.org/>.
- [86] Site de GCJ. <http://gcc.gnu.org/java/>.
- [87] SWT : The standard widget toolkit. <http://www.eclipse.org/swt/>.
- [88] Site d'azureus. <http://azureus.sourceforge.net/>.
- [89] Site de jEdit. <http://www.jedit.org/>.
- [90] Site de JFreeChart. <http://www.jfree.org/jfreechart/>.
- [91] Site du projet Harmony. <http://incubator.apache.org/harmony/>.
- [92] Commentaire de Dalibor Topic (développeur kaffe et classpath) au sujet de l'absence d'accord entre la fondation apache et classpath. [http://www.osnews.com/permalink.php?news\\_id=14198&comment\\_id=110641](http://www.osnews.com/permalink.php?news_id=14198&comment_id=110641), April 2006. OsNews.
- [93] Mark Wielaard. Toward a free java. <http://lwn.net/Articles/184967/>, May 2006. LWN.
- [94] Martin LaMonica. Ibm urges sun to make java open source. [http://news.com.com/IBM+urges+Sun+to+make+Java+open+source/2100-1007\\_3-%5165427.html](http://news.com.com/IBM+urges+Sun+to+make+Java+open+source/2100-1007_3-%5165427.html), February 2004. CNET News.com.
- [95] Eric S. Raymond. Open letter to sun : Let java go. <http://www.catb.org/~esr/writings/let-java-go.html>, February 2004.
- [96] Martin LaMonica. Sun reluctant to make java open source. [http://news.com.com/Sun+reluctant+to+make+Java+open+source/2100-7344\\_3-%5173427.html](http://news.com.com/Sun+reluctant+to+make+Java+open+source/2100-7344_3-%5173427.html), March 2004. CNET News.com.
- [97] Martin LaMonica. Door to java source code opens a crack wider. [http://news.com.com/Door+to+Java+source+code+opens+a+crack+wider/2100-7%344\\_3-5621235.html](http://news.com.com/Door+to+Java+source+code+opens+a+crack+wider/2100-7%344_3-5621235.html), 2005. CNET News.com.
- [98] Dalibor Topic. Welcome to the java melodrama show! <http://www.advogato.org/person/robilad/diary.html?start=65>, March 2005.

- [99] Andrew Orlowski. 'get a lawyer!' sun tells developers. [http://www.channelregister.co.uk/2005/03/17/two\\_new\\_java\\_licenses/](http://www.channelregister.co.uk/2005/03/17/two_new_java_licenses/), March 2005. Channel Register.
- [100] Martin LaMonica. Sun to make java more linux-friendly. [http://news.com.com/Sun+to+make+Java+more+Linux-friendly/2100-7344\\_3-60%68852.html](http://news.com.com/Sun+to+make+Java+more+Linux-friendly/2100-7344_3-60%68852.html), May 2006. CNET News.com.
- [101] Joris Evers. Sun promises to open-source java. [http://news.com.com/Sun+promises+to+open-source+Java/2100-7344\\_3-607276%0.html](http://news.com.com/Sun+promises+to+open-source+Java/2100-7344_3-607276%0.html), May 2006. CNET News.com.
- [102] Stephen Shankland. Sun expands open-source java plan. [http://news.com.com/Sun+expands+open-source+Java+plan/2100-7252\\_3-61056%01.html](http://news.com.com/Sun+expands+open-source+Java+plan/2100-7252_3-61056%01.html), August 2006. CNET News.com.
- [103] Les plans de sun pour la libération de java. <https://jdk.dev.java.net/news-08-14-06.html>.
- [104] Site *Open Sourcing the JDK*. <http://community.java.net/jdk/opensource/>.
- [105] Tom Tromey. Open source java. <http://tromey.com/blog/?p=262>, August 2006.
- [106] 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.
- [107] 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.
- [108] Jonathan Skillings. Kodak sues sun over aspects of java. <http://news.com.com/2100-1001-836322.html>, February 2002. CNET News.com.
- [109] Pamela Jones. Kodak wins Java lawsuit against Sun. <http://www.groklaw.net/article.php?story=20041003041632172>, October 2004.
- [110] Stephen Shankland. Sun settles Kodak's Java suit for \$92 million. [http://news.com.com/2100-1012\\_3-5401804.html](http://news.com.com/2100-1012_3-5401804.html), October 2004. CNET News.com.
- [111] Charles Babcock and John Foley. The cost of ideas. <http://www.informationweek.com/story/showArticle.jhtml?articleID=499005%78>, October 2004. Information Week.
- [112] 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.
- [113] Elijah Newren. New module decisions for 2.16. <http://mail.gnome.org/archives/devel-announce-list/2006-August/msg00000%.html>, August 2006.
- [114] Floyd Marinescu. Red hat sued over hibernate 3 orm patent infringement claim. <http://www.infoq.com/news/RedHat-Sued-Due-to-Hibernate-3-0>, June 2006.
- [115] Site d'hibernate. <http://hibernate.org/>.
- [116] Bruce Perens. The monster arrives : Software patent lawsuits against open source developers. <http://technocrat.net/d/2006/6/30/5032>, June 2006.