



Droit des logiciels

Tout ce que vous avez toujours voulu savoir sans oser le demander...

François Pellegrini

EQUIPE PROJET

BACCHUS

Bordeaux

Sud-Ouest

14/02/2013

Droit et informatique

- Droit et informatique sont des disciplines de même nature
 - Elles consistent à construire, à partir de quelques éléments de base, les règles de fonctionnement d'univers complets
- Le problème, c'est que ces univers sont l'univers dans lequel nous vivons !
 - Les générations précédentes vivaient dans le « monde de Kafka »
 - Notre génération vit également dans le « monde de Matrix »
- Intérêt d'une compréhension mutuelle entre informaticiens et juristes

Mon travail...

- Faire comprendre l'informatique aux juristes...
 - Et faire comprendre le droit aux informaticiens !
- Revenir aux fondements de l'informatique pour questionner les fondements du droit
 - Ainsi que leurs pratiques

Émergence du logiciel comme objet juridique

Apparition du logiciel

- Les premiers calculateurs ne possédaient pas de logiciel
 - Programmés « à la main »
- Le logiciel (et l'ordinateur) sont apparus avec l'idée que le programme pouvait être stocké dans la même mémoire que celle utilisée pour les données qu'il manipule
 - Architecture dite « de Von Neumann »
 - Le logiciel est une donnée (presque) comme une autre !
 - Possibilité de programmes auto-modifiables
 - Méta-programmation

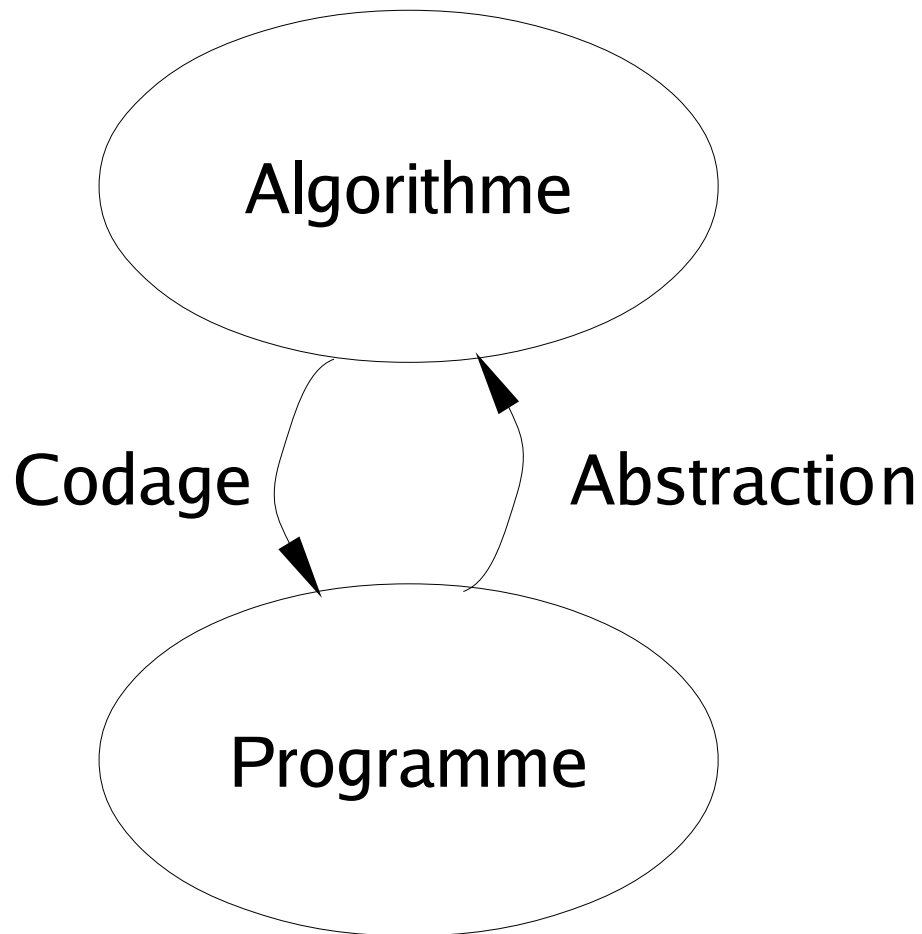
Apparition du droit du logiciel (1)

- Tant que les logiciels étaient spécifiques à un ordinateur donné, la question de leur statut ne se posait pas
 - Les logiciels étaient fournis « gratuitement » aux utilisateurs, en tant que « fournitures annexes »
 - Au même titre que les manuels d'utilisation
- Les fournisseurs encourageaient leurs clients à modifier les logiciels et à partager les modifications
 - Soutien à la création de « clubs d'utilisateurs » servant à l'échange de ces améliorations
 - Mutualisation et donc réduction des coûts de maintenance logicielle
 - Avantage compétitif sur les concurrents
 - Les principes du logiciel libre avant l'heure !

Apparition du droit du logiciel (2)

- La question du statut du logiciel s'est posée lorsque sont apparus les premiers ordinateurs compatibles avec les grands systèmes IBM, à la fin des années 1960
 - Les fournisseurs de matériels compatibles utilisaient les logiciels d'IBM sans en payer le prix ni même le coût
- IBM décida donc de facturer séparément logiciel et matériel
 - Politique de « dégroupage » (« *unbundling* »)
 - Mais comment caractériser juridiquement le logiciel ?

Nature du logiciel



- Les algorithmes sont :
 - Des idées
 - Des mathématiques
- Les programmes sont :
 - Des œuvres de l'esprit
 - Du discours
 - Humain ↔ humain
 - Humain → ordinateur
 - Des processus, lorsqu'ils sont exécutés

- Similaire au processus de création littéraire

Quel statut pour les langages ?

- Un langage n'est pas une œuvre
 - Il permet d'écrire des œuvres
 - Il est d'un niveau d'abstraction supérieur
 - Pas de revendication possible sous le régime du droit d'auteur
- Un langage informatique est aussi une langue de communication humaine
 - Permet l'échange d'informations entre humains
- Les langages informatiques ne peuvent donc être appropriés sans porter préjudice à des droits plus élevés
- Un format de fichiers définit la grammaire d'un langage
 - Accorder des brevets sur les formats de fichiers reviendrait donc à pouvoir monopoliser un langage

Quel statut pour les logiciels ?

- Trois voies offertes au législateur :
 - Brevet
 - Instrument inadapté : portée, coûts de transaction, etc.
 - Mais certains veulent y revenir...
 - Droit *sui generis*
 - Durée d'établissement des droits nationaux et conventions internationales
 - Droit d'auteur
 - Problèmes à la marge mais applicabilité immédiate
 - Convention de Berne de 1886 déjà signée par plus de 170 pays

Droit d'auteur adapté

- Par son rattachement au droit d'auteur, le logiciel est assimilé à une œuvre de l'esprit
 - En France, loi du 3 juillet 1985
 - Directive européenne 91/250/CE (1991)
 - Article 10 des accords ADPIC (1994)
 - Article 4 du traité OMPI WCT (1996)
- Le logiciel est cependant aussi un produit substituable voué à rendre un service
 - Adaptation du droit d'auteur (« droit d'auteur adapté »)
 - Question de la garantie

http://www.wto.org/french/docs_f/legal_f/27-trips_04_f.htm#droit
<http://www.wipo.int/documents/fr/diplconf/distrib/94dc.htm>

Adaptation des droits patrimoniaux (1)

- Ajout de nouvelles exceptions
 - « Actes nécessaires pour permettre l'utilisation du logiciel »
 - « Observation du fonctionnement »
 - En fait, simple rappel de la loi
 - Copie de sauvegarde
 - Si aucun autre moyen fourni par l'éditeur
 - La décompilation
 - Interdite sauf « à fin d'interopérabilité »
- Suppression de l'exception de copie privée
 - Nécessite une licence pour chaque exemplaire du logiciel

Adaptation des droits patrimoniaux (2)

- Pour les auteurs salariés ou fonctionnaires, transfert automatique de la titularité des droits patrimoniaux à l'employeur
 - C'est l'employeur qui décide de la vie de l'œuvre :
 - Divulcation, choix de la licence, etc.
 - Apparition d'un statut d'« auteur prolétaire »
- Les auteurs non salariés restent titulaires des droits patrimoniaux sur leurs œuvres logicielles
 - Nécessité de transférer les droits des sous-traitants
 - Problème des stagiaires non rémunérés

Adaptation des droits extra-patrimoniaux

- Amoindrissement des droits extra-patrimoniaux
 - L'auteur salarié ou fonctionnaire ne peut s'opposer à la modification de l'œuvre
 - Reste le droit au nom...
 - Plus courant dans le monde du jeu vidéo que de la comptabilité !

Compilation, décompilation et interopérabilité

Langages de bas et haut niveau

- Les ordinateurs ne comprennent que le « langage machine »
 - Instructions très rudimentaires
 - Spécifique à un processeur donné
 - Et à un système d'exploitation donné
 - Code très peu expressif
 - Risques de bogues élevé
 - Dépend du nombre de lignes que l'on écrit
- Besoin d'écrire des programmes dans des langages plus expressifs
 - Meilleure compréhension du code
 - Indépendance vis-à-vis du type de processeur
 - Meilleures maintenabilité et portabilité
 - Coût économique bien inférieur

Compilation et décompilation

- La compilation consiste à traduire un programme écrit dans un langage de haut niveau en un langage de bas niveau, susceptible d'être exécuté par un ordinateur
 - Le « code source » est le programme écrit en langage de haut niveau que l'on veut traduire
 - La « forme préférée » d'écriture d'un programme
 - Le « code objet » est le programme résultant écrit dans un langage de bas niveau
 - Définit le « programme exécutable »
- On appelle « décompilation » l'action inverse
 - Bien plus difficile à mettre en œuvre
 - Informations structurelles de haut niveau « diluées » dans le code objet

Traduction automatique

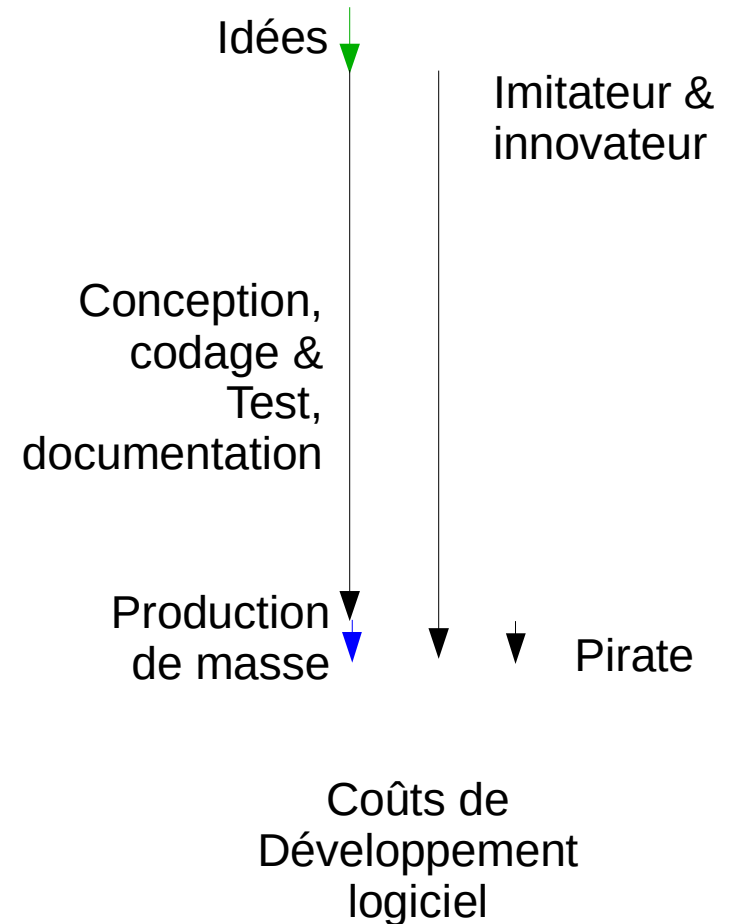
- Du fait que la traduction du code source en code objet se fait de façon automatique, le code objet est une œuvre intégralement dérivée du code source, sans aucun apport original d'un autre auteur
 - Le créateur d'un outil n'a aucun droit sur les créations réalisées au moyen de cet outil
 - Pensez aux dictionnaires, pinceaux, etc...
 - Cas particulier : inclusion de fragments de code du compilateur dans le programme exécutable
 - Hypothèse à revoir quand existeront des intelligences artificielles
 - « Valladolid 2.0 » !

Interdiction de la décompilation

- Compiler et décompiler crée des œuvres dérivées de l'œuvre originale
- Ces copies ne peuvent donc être exploitées qu'avec l'accord de l'ayant droit des œuvres originales
- La décompilation est donc interdite, sauf permission explicitement donnée par l'ayant-droit

Modèle économique du logiciel

- L'observation des fonctionnalités d'un logiciel existant permet à un nouvel arrivant de réaliser un logiciel concurrent
- Il n'y a concurrence libre et non faussée que s'il paye les mêmes coûts d'entrée sur le marché en re-codant son logiciel
 - Donc il ne peut réutiliser le code existant, par décompilation et/ou copie servile
- Pendant ce temps, l'innovateur original peut progresser
 - S'il ne le fait pas, il sera dépassé par les innovations des autres



Interopérabilité

- Les formats de fichiers créent des marchés captifs
 - La vraie valeur est dans les données !
- Possibilité de décompiler « à fin d'interopérabilité »
 - Afin de garantir l'existence de la concurrence libre et non faussée
 - Lutter contre les effets de réseau
 - Strictement encadrée par la loi

Statut des interfaces (1)

- La libre concurrence suppose qu'on puisse interfacier un nouveau logiciel avec un logiciel existant
 - Voire le remplacer en s'interfaçant avec les logiciels tiers avec lesquels ce logiciel interagissait
- Or, lorsqu'on veut s'interfacier avec un logiciel existant, il faut respecter ses interfaces
 - Décrites dans des fichiers d'en-tête, des fichiers de classes, etc
- Pour s'interfacier, il faut donc « recopier » le format des interfaces
 - Est-on un contrefacteur au sens du droit d'auteur ?

Statut des interfaces (2)

- La libre concurrence suppose qu'on puisse interfacier un nouveau logiciel avec un logiciel existant
 - Voire le remplacer en s'interfaçant avec les logiciels tiers avec lesquels ce logiciel interagissait
- Or, lorsqu'on veut s'interfacier avec un logiciel existant, il faut respecter ses interfaces
 - Décrites dans des fichiers d'en-tête, des fichiers de classes, etc
- Pour s'interfacier, il faut donc « recopier » le format des interfaces
 - Est-on un contrefacteur au sens du droit d'auteur ?

Licences

Licence

- La licence est une offre de contrat de la part du fournisseur, qui définit les conditions d'utilisation d'une œuvre
 - Le terme juridique exact est : « pollicitation »
- Basée sur les droits d'auteur ou le copyright
 - Convention de Berne de 1886
- Classiquement, une licence limite les droits d'usage d'une œuvre :
 - Interdiction de diffusion publique
 - Interdiction de reproduction, même partielle
 - ...

Licences de logiciels (1)

- Il existe de nombreux types de licences de logiciels :
 - « Logiciel propriétaire » (« privatif ») : tous les droits sont réservés par leur titulaire
 - Cas de la majorité des logiciels du commerce
 - Dans la quasi totalité des cas, pour ces logiciels :
 - Le client n'est propriétaire que du support, et pas du logiciel qu'il contient
 - Le fournisseur dégage toute responsabilité en cas de vices cachés (bogues)
 - Le fournisseur peut arrêter la maintenance du logiciel à tout moment

Licences de logiciels (2)

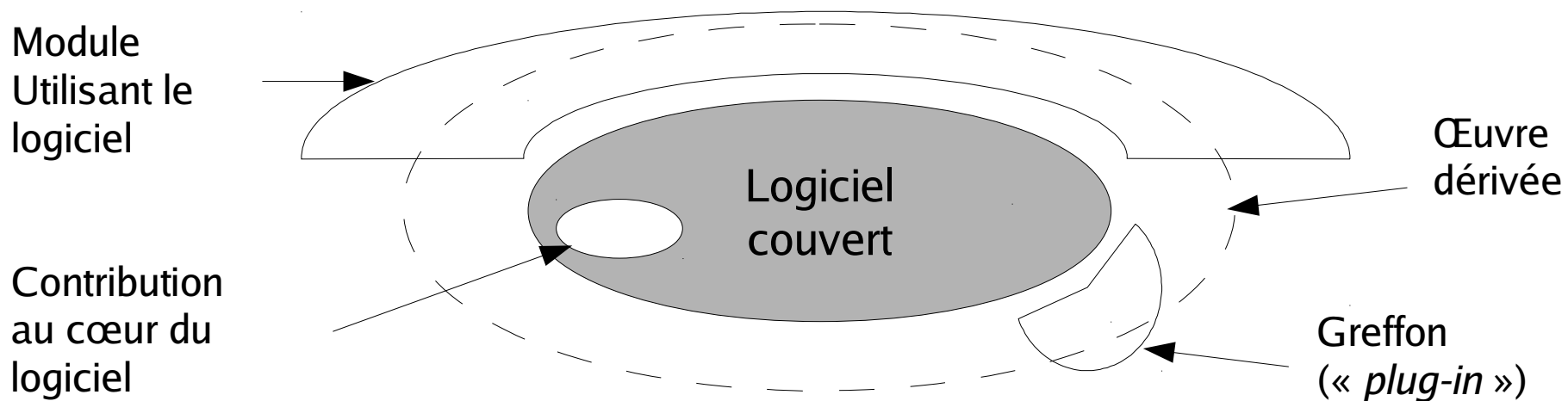
- « Partagiciel » (« *Shareware* ») : logiciel privatif diffusable gratuitement mais pour lequel une contribution est demandée au bout d'une période d'essai
 - Logiciel privatif où seul diffère le mode de distribution
- « Gratuiticiel » (« *Freeware* ») : logiciel privatif gratuit mais ne donnant pas nécessairement d'autres droits
 - Parfois pas même celui de redistribution
- « Logiciel libre » (« *Free software* ») : logiciel donnant de nombreux droits à ses utilisateurs
 - N'est **pas** équivalent à un « *freeware* » !
 - Pas nécessairement gratuit
- « De domaine public »

Les licences libres (1)

- Ont en commun les « quatre libertés »
 - Liberté d'utilisation pour tout usage
 - Liberté de copie du logiciel original
 - Liberté de modification du code source
 - Implique la liberté de consultation
 - Liberté de redistribution du logiciel modifié
- Les différences entre licences libres portent sur les modalités de redistribution du logiciel modifié :
 - Si persistance des obligations de redistribution du code source : « *copyleft* »
 - Si absence de cette obligation : « *non copyleft* »

Les licences libres (2)

- Trois principaux types de licences libres :
 - Licences « persistantes »
 - Licences « évanescentes »
 - Licences « diffusives »
- Les licences s'appliquent à trois entités, le plus souvent au travers de la notion d'œuvre dérivée :



Conclusion

Droit mouvant et encore à construire

- Interaction avec de nombreux autres droits
 - Droit des « mesures techniques de protection »
 - Droit des brevets
 - ...
- Sujet passionnant !