



**HAL**  
open science

# L'originalité des œuvres logicielles créées par composition de briques préexistantes

François Pellegrini

► **To cite this version:**

François Pellegrini. L'originalité des œuvres logicielles créées par composition de briques préexistantes. L'originalité du logiciel en question, CNEJITA; AFDIT, Apr 2021, Paris / Virtual, France. hal-03202438

**HAL Id: hal-03202438**

**<https://inria.hal.science/hal-03202438>**

Submitted on 20 Apr 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License

---

# L'originalité des œuvres logicielles créées par composition de briques préexistantes

François Pellegrini  
Professeur, Université de Bordeaux  
francois.pellegrini@u-bordeaux.fr

Ce document est copiable et distribuable librement et gratuitement à la condition expresse que son contenu ne soit modifié en aucune façon, et en particulier que le nom de son auteur et de son institution d'origine continuent à y figurer, de même que le présent texte.

# Le logiciel

Une création artisanale et artistique  
exploitée de façon industrielle

# Logiciel (1)

---

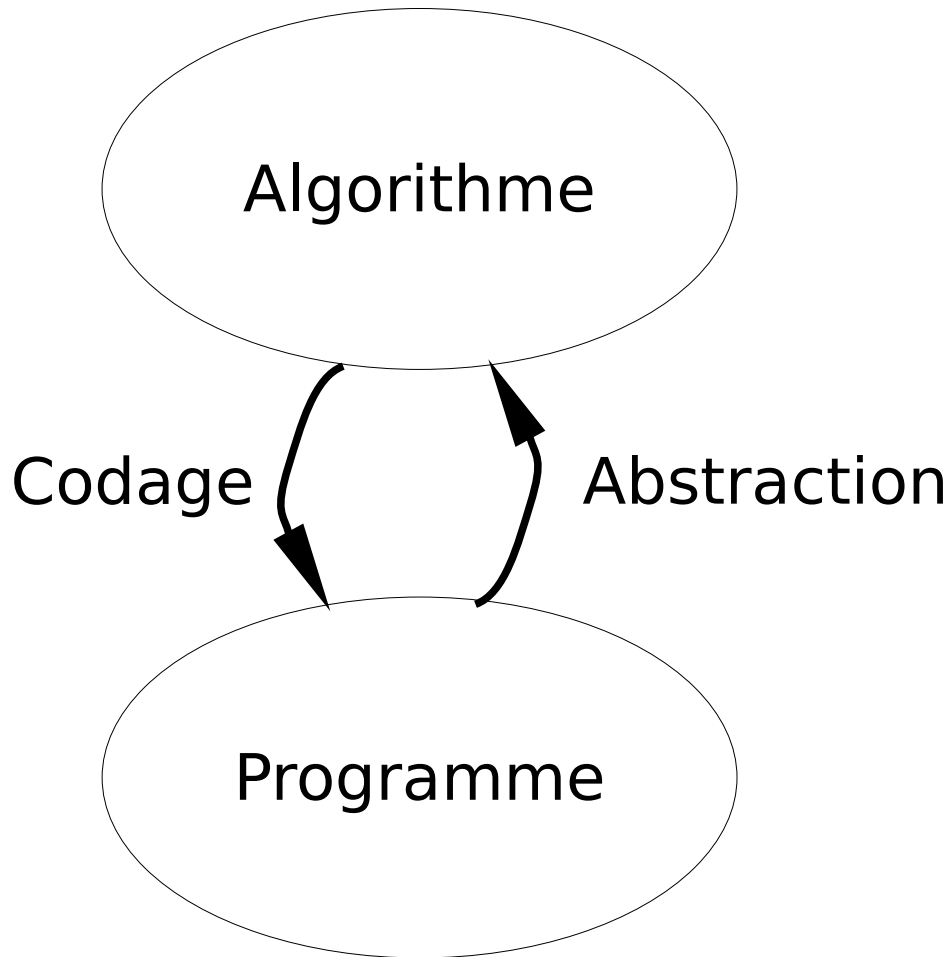
- « Expression, dans un ou plusieurs langages de programmation, d'un ensemble d'algorithmes visant à la réalisation d'une tâche intellectuelle donnée »
  - Habituellement exprimé sous forme de textes
    - Peut toujours se ramener à une forme textuelle
- Terme exprimant la nature « logique » du programme informatique
  - Bien immatériel

# Logiciel (2)

---

- Un logiciel est caractérisé par :
  - Sa forme programmée
    - Le code source, résultat du travail créatif du programmeur
    - Le code objet, résultat de la traduction automatique du code source par d'autres logiciels
      - Tel que le code exécutable en langage machine
  - Sa forme communiquée
    - Ce que perçoit l'utilisateur
    - Discernable à travers les fonctionnalités exprimées par le code en cours d'exécution sur un système informatique
    - La « caverne de Platon » du code

# Algorithmes et programmes



- Algorithmes :
  - Des idées
  - Des mathématiques
- Programmes :
  - Des œuvres de l'esprit
  - Du discours
    - Humain → humain
    - Humain → ordinateur
- Traitements :
  - Lorsque exécutés

- Analogue au processus de création littéraire

# Le droit du logiciel

Une adaptation du droit d'auteur à cette  
catégorie d'œuvre particulière

# L'œuvre en droit d'auteur

---

- L'œuvre est une création de forme
  - C'est la forme qui sera protégée, et non les idées et les concepts
- « Les idées sont de libre parcours » !  
(H. Desbois)



# Critère de protection

---

- L'œuvre doit être « originale »
  - Au sens du droit d'auteur !
    - Attention au mésusage du terme « originalité »
      - Par exemple : « original vs. copie »
    - Reflet de la « personnalité de l'auteur »
- L'acte créatif de l'auteur s'exprime par le chemin qu'il choisit de prendre dans un espace de liberté formelle lui permettant d'exprimer sa personnalité
  - Deux portraits de la même personne par deux artistes différents n'expriment pas les mêmes choix « artistiques » (ni choix de techniques)

# Créations échappant à la protection

---

- Une création de forme ne reflétant pas la personnalité de son concepteur ne sera pas éligible à la protection par le droit d'auteur
  - Tables mathématiques, annuaires des marées, etc.
- Cependant, leur présentation graphique pourra être éligible si elle reflète la personnalité de son auteur
  - Logos, décorations, etc.

# Quelle preuve de l'originalité ?

---

- Il est plus facile d'exprimer la question sous la forme négative :
  - « Une création n'est pas originale si son créateur ne dispose d'aucun espace de liberté formelle »
- S'il existe un espace de liberté formelle, alors il y a présomption d'originalité
  - Comment l'auteur peut-il justifier son style ?
- Bien distinguer entre la question de l'originalité et celle du plagiat !
  - Un plagiat est une œuvre composite

# Droit d'auteur adapté

---

- Par son rattachement au droit d'auteur, le logiciel est assimilé à une œuvre de l'esprit
  - Loi du 3 juillet 1985
  - Directive européenne 91/250/CE (1991)
    - Transposée notamment dans l'art. L.122-6-1 CPI
- Le logiciel est cependant aussi un produit substituable voué à rendre un service
  - Adaptation du droit d'auteur
    - D'où le terme de « droit d'auteur adapté »
    - Alors que les éléments graphiques et sonores restent couverts par le droit d'auteur « classique »

# L'originalité des œuvres logicielles (1)

- L'auteur de logiciel est un auteur comme un autre
  - Le droit ne distingue pas ni ne crée des obligations plus strictes pour ces auteurs
- Sa liberté de choix s'exprime pleinement dans son style d'écriture
  - Liberté formelle d'écrire du code, même erroné
- La notion de « logique contraignante » relève de l'algorithmique, pas de la forme
  - Question des interfaces de programmation

# L'originalité des œuvres logicielles (2)

- Parfois reformulée sous le terme d'« apport intellectuel » dans le cas des œuvres logicielles
- Reformulation totalement inopportune
  - Méconnaissance de la réalité du travail de l'auteur informaticien
  - Critère trop vague
    - Relève tant des fonctionnalités que de la forme
    - Risque de s'attacher à la forme communiquée plutôt qu'à la forme programmée
  - On ne doit pas distinguer là où la loi ne distingue pas !

# L'originalité des œuvres logicielles (3)

---

- Art. 1 al. 3 de la directive 91/250/CE :  
« Un programme d'ordinateur est protégé s'il est original, en ce sens qu'il est la création intellectuelle propre à son auteur. Aucun autre critère ne s'applique pour déterminer s'il peut bénéficier d'une protection. »

# L'originalité des œuvres logicielles (4)

- Du fait de la liberté formelle laissée à l'auteur dans la réalisation du code source de son programme, il ne peut exister en pratique de logiciel non original
  - Espace de liberté dans la spécification formelle des traitements à réaliser
    - Choix et organisation formelle des structures de contrôle
    - Noms de variables/registres, fonctions, fichiers
      - À part ceux des interfaces de programmation (API)
  - Liberté absolue d'écriture des commentaires
  - Liberté absolue d'écriture des matériels de conception préparatoire



# Critères non pertinents (1)

---

- La « nouveauté »
  - « Encore une histoire de mousquetaires ?! »
  - Des navigateurs web différents sont bien chacun des œuvres originales
    - Alors qu'ils doivent respecter des normes de comportement très contraignantes
- La « nouveauté » est un concept lié aux fonctionnalités mises en œuvre dans le cadre de la forme communiquée, et pas à l'écriture de la forme programmée
  - Critère pour le droit des brevets, pas d'auteur
    - Le terme « propriété intellectuelle » mélange tout

# Critères non pertinents (2)

---

- Le « mérite »
  - Qui se souvient encore des peintres « officiels » de la fin du XIXe siècle, par rapport aux « refusés » ?
- Pente glissante lorsqu'on manipule certains concepts inadéquats qui s'apparentent à un jugement « au poids » :
  - « Apport intellectuel »
  - « Banalité »

# L'œuvre réalisée à plusieurs (1)

---

- Trois catégories juridiques (L.113-2 CPI)
  - Œuvre de collaboration
    - Œuvre à la réalisation de laquelle ont concouru plusieurs personnes physiques
    - Droits partagés
  - Œuvre composite (aussi appelée « œuvre dérivée »)
    - Œuvre nouvelle à laquelle est incorporée une œuvre préexistante sans la collaboration des auteurs de celle-ci
    - Exploitable sans préjudice des droits attachés aux œuvres incorporées

# L'œuvre réalisée à plusieurs (2)

---

- Œuvre collective
  - Œuvre créée à l'initiative d'une personne physique ou morale qui l'édite, dans laquelle la contribution des différents auteurs se fond [...]
  - Exercice des droits patrimoniaux par la personne éditrice

# Génération procédurale (1)

---

- La génération procédurale est une pratique courante en informatique
  - Compilation d'un code source en code objet
- L'usage d'un outil ne crée pas de droits de l'auteur de l'outil sur l'œuvre dérivée produite
  - Application de procédures de transformation totalement automatisées
  - Le code résultant n'est pas une œuvre composite
    - Sauf cas d'inclusion de fragments de code explicitement écrits par l'auteur de l'outil

# Génération procédurale (2)

- Bien distinguer les caractérisations juridiques de :
  - La forme programmée
    - Encadre les actions possibles de l'utilisateur
  - La forme communiquée, intégralement décrite en termes fonctionnels par la forme programmée
    - Y compris dans le cas de logiciels faisant appel au « hasard »
    - Comportement effectif décrit par le paramétrage
  - Les créations pouvant résulter de l'interaction d'un utilisateur avec la forme communiquée
    - Éventuellement considérées comme des œuvres

---

# Licences

A licence to kill... ?

# Licence (1)

---

- 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 le droit d'auteur ou le copyright
  - Convention de Berne de 1886
- Classiquement, une licence ne concède que les droits d'usage minimaux :
  - Interdiction de diffusion publique
  - Interdiction de reproduction, même partielle
  - ...



# Licence (2)

---

- Ne pas confondre « licence » et « *license* »
- Aux États-Unis, la « *license* » est un texte relatif au droit d'auteur
  - De niveau fédéral, d'interprétation uniforme
  - À la différence des « contracts », par État
- En droit français, une « licence » est un contrat qui permet de d'organiser l'application du droit
  - Par exemple : licence de marque
  - À rapprocher des contrats d'adhésion (CGU)

# Logiciel libre

---

- Le « logiciel libre » est une innovation juridique et non pas technique
  - S'appuie sur le mécanisme du droit d'auteur applicable aux logiciels
  - Implique des droits et des devoirs
- Favorise l'émergence de modèles économiques déconcentrés adaptés à l'économie des biens immatériels
  - Permet la micro-crédation de valeur ajoutée

# Licences libres (1)

---

- Ont en commun les « quatre libertés »
  - Liberté d'utilisation pour tout usage
  - Liberté de copie du logiciel obtenu
  - Liberté de modification du logiciel
    - Nécessite l'accès au code source
  - Liberté de redistribution du logiciel modifié
    - Capitalisation du savoir
    - Mutualisation des développements

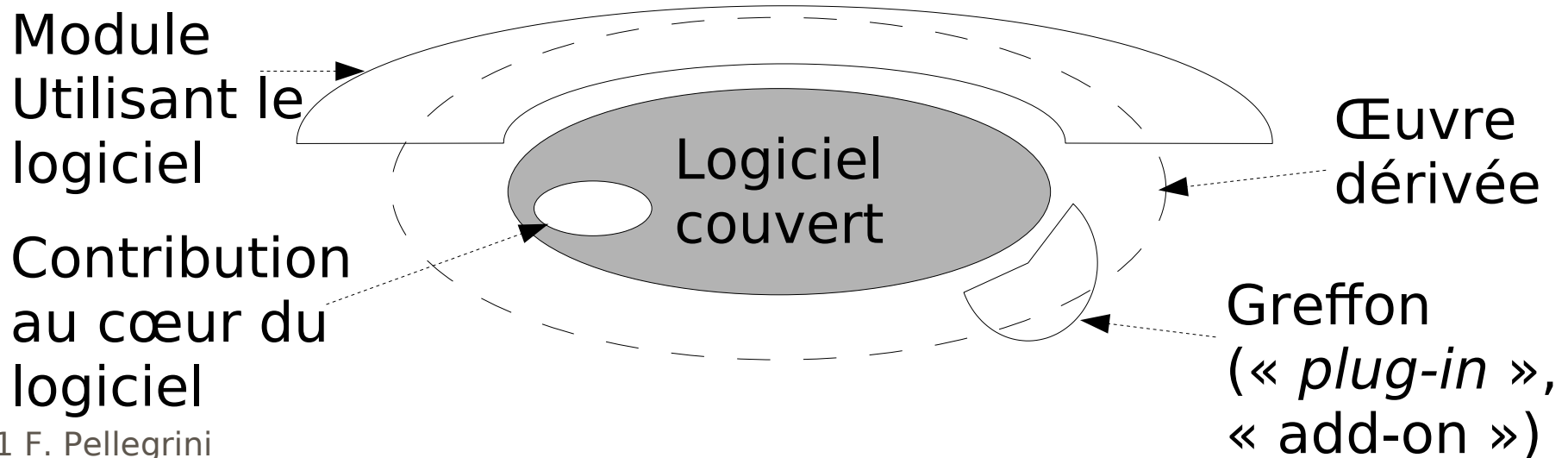
# Licences libres (2)

---

- Les différences entre licences libres portent sur les modalités de redistribution du logiciel modifié :
  - Si persistance des obligations de redistribution du code source : « *copyleft* »
    - Liberté collective
  - Si absence de cette obligation : « *non copyleft* »
    - Liberté individuelle
- Caractérisent des modèles économiques différents

# Mode d'action des licences libres

- Trois types principaux d'interaction avec un module logiciel :
  - Modification / Extension (greffon) / Utilisation
- Création d'une œuvre composite dont le sort est régi par les termes de la licence couvrant le(s) module(s)



# Contrefaçon de logiciel sous licence libre

- Comme toute licence imposant des obligations, les licences libres peuvent faire l'objet de contrefaçon
  - Notamment violation des clauses « *copyleft* »
    - Voir notamment : [gpl-violations.org](http://gpl-violations.org)
- La violation des termes de la licence entraîne :
  - La terminaison de la licence
    - Aspect contractuel intégré à la licence elle-même
    - Procédure de remédiation automatique
    - Sécurisation des tiers destinataires
  - Des risques de poursuite en contrefaçon

# Et l'originalité dans tout ça ?

Au travers de quelques études de cas...

# Composition de logiciels

---

- La personne assemblant plusieurs briques logicielles crée une œuvre composite
  - Exploitable sans préjudice des droits des ayants droit des œuvres initiales
    - Se référer au texte de la licence couvrant chacun des modules que l'on souhaite intégrer
      - Problématique d'incompatibilités juridiques entre licences
- L'originalité de l'auteur s'exprime dans l'organisation formelle de la composition
  - Arrêt « Paradis », CC, Civ. 1<sup>re</sup>, 06-19.021, n° 1108, 13/11/2008



# En cas de litige... (1)

---

- L'existence des droits ne transparait qu'à travers les preuves que l'on peut exhiber aux yeux du juge
- Il y a litige s'il existe deux parties se disputant au sujet d'un ou plusieurs logiciels

# En cas de litige... (2)

---

- En termes de droit d'auteur, à partir de quel moment peut-on caractériser la contrefaçon ?
- Ne pas confondre les questions d'originalité et de titularité des droits !
- Un plagiat est une œuvre originale
  - Les éléments incorporés sont des œuvres, originales en elles-mêmes
  - L'action du plagiaire crée une œuvre composite du fait de son apport propre
    - Le problème est qu'il méconnaît les droits des tiers

# Similarité des fonctionnalités ?

---

- A accuse B d'avoir contrefait son logiciel parce que  $L_B$  fait la même chose que  $L_A$ 
  - Liberté d'observation des fonctionnalités
    - Art. L.122-6-1 III CPI
  - Liberté de refaire un logiciel réalisant les mêmes actions
    - Y compris droit à la décompilation à fin d'interopérabilité pour briser les marchés captifs autour des formats de données et/ou d'échange fermés
- Moyen invalide

# Similarité du code ? (1)

---

- A accuse B d'avoir contrefait son logiciel parce que le code de  $L_B$  ressemble « beaucoup » à celui de  $L_A$ 
  - Existence d'une source commune ?
    - Notamment une brique tierce sous licence libre
  - Existence d'une fuite de données ?
    - Espionnage industriel ? Salarié indélicat ? Fuite accidentelle ?
  - Respect d'interfaces ?
    - Existence d'un espace de liberté formelle ?

## Similarité du code ? (2)

---

- Ce n'est pas parce que le code de  $L_B$  diffère de celui de  $L_A$  qu'il n'y a pas contrefaçon !
  - Le code de  $L_B$  a pu être obtenu par traduction (semi-)automatique à partir de celui de  $L_A$
- Rôle de l'expert dans la détermination du degré de fortuité des similarités d'expression du code

# Détermination de l'originalité (1)

---

- En tout état de cause, l'originalité d'une œuvre logicielle peut/doit toujours être présumée
  - L'auteur a juste « fait son travail »
  - L'équivalence des comportements observés n'est pas significative
- Aucun intérêt économique à dénier l'originalité à certains logiciels
  - Base juridique incertaine
    - Laisse les auteurs sans protection
    - Jugement au « mérite »
  - Non pertinent vis-à-vis des actions à réprimer

# Détermination de l'originalité (2)

---

- Les litiges en matière de droits d'auteur doivent se résoudre à travers l'étude du code source et de la façon dont il a été produit
  - Analyse qualitative et non pas quantitative
    - Pas de « *sweat of the brow* »
  - L'existence d'une contribution, même minimale, est suffisante
    - Marcel Duchamp et sa « L.H.O.O.Q. »

# Détermination de l'originalité (3)

---

- Pas de renversement de la charge de la preuve
  - C'est au plaignant de faire la preuve qu'un code a été obtenu de façon déloyale, en violation du modèle économique de la création logicielle :
    - Copie servile, traduction automatique, etc.



# Conclusion

---

- L'originalité s'exprime dans l'écriture de textes en langages informatique
  - À l'image de l'écriture littéraire « classique »
- Risque économique majeur dans le cas où l'on méconnaît l'originalité de ces œuvres
  - Décisions jurisprudentielles *contra legem*
  - Mais obligation de « faire avec »...
- Nécessité de traiter les litiges sur plusieurs fronts simultanés :
  - Droit d'auteur (mais danger !)
  - Aspects contractuels (actions non exclusives)