

UNIVERSITÉ PARIS-SUD

Faculté Jean Monnet – Droit, Économie, Gestion

Année universitaire 2012-2013

Valeur et protection des langages de programmation.

Mémoire de recherche.
Master 2 recherche mention Droit Innovation Communication Culture
spécialité Droit des Nouvelles Propriétés.

Présenté par :
Jean-Baptiste VAROQUEAUX

Sous la direction du professeur :
Pierre SIRINELLI

Version en ligne sur <http://varoqueaux.awardspace.biz> le 6 juin 2013.
Tous droits réservés.

« L'université n'entend donner ni approbation, ni improbation aux opinions contenues dans ce mémoire lesquelles doivent être considérées comme propres à l'auteur. »

Remerciements

Outre mon directeur de mémoire, monsieur le professeur Pierre Sirinelli, et ma directrice de master, madame Alexandra Bensamoun, qui ont nourri ma réflexion et m'ont laissé libre de mener cette étude qui me tenait à cœur jusqu'à son terme, je souhaite remercier monsieur Franck Macrez pour ses conseils avisés ainsi que Julien Lorent et Bastien Jorge pour leur soutien technique.

Sommaire*

Remerciements

Introduction

Chapitre 1 :La valeur considérable des langages de programmation

- I)L'opportunité économique et technique favorable à la protection des langages
- II)La protection principale des langages de programmation par le droit d'auteur

Chapitre 2 :Le régime juridique complexe des langages de programmation

- I)D'un régime distributif vers un régime unitaire préférable
- II)Une exploitation multiforme et raisonnée des langages de programmation

Conclusion

Sources et bibliographie

Table des matières

* Table des matières en fin de document

Introduction

1. – Algorithmes, langages de programmation et programmes d'ordinateur, voici un trio déprécié par les spécialistes du droit d'auteur qui, pourtant, est à la fois le fleuron de notre science et un formidable outil d'expression artistique. L'informatique connaît des applications monumentales et d'autres plus fonctionnelles ; cela va des logiciels ludiques aux grands simulateurs d'explosion nucléaire, des programmes utiles – comme les outils de bureautique – aux indispensables – comme les systèmes des laboratoires de recherche. Il est possible d'interpréter des données satellites ou de corriger l'imperfection du matériel, allant de la correction des yeux rouges sur une photographie à celle des perturbations atmosphériques pour les télescopes géants. Les programmes informatiques sont donc une solution technique indéniablement révolutionnaire, pour autant leur conception suppose de recourir à des algorithmes et des langages de programmation, de telle manière qu'en pratique ces trois éléments sont indissociables.

2. – Le langage est une « fonction d'expression de la pensée et de communication entre les hommes, mise en œuvre au moyen d'un système de signes vocaux et éventuellement de signes graphiques qui constitue une langue »¹, il se différencie de la langue qui est une « utilisation individuelle du langage, façon personnelle de s'exprimer par le langage »², en d'autres mots le dialecte voire le style de celui qui emploie un langage. La programmation est une « élaboration et codification de la suite d'opérations formant un programme »³ lequel est un « ensemble d'instructions, rédigé dans un langage de programmation, permettant à un système informatique d'exécuter une tâche donnée »⁴, soit un synonyme de logiciel. Enfin, le langage de programmation est un « ensemble codé de signes utilisé pour la programmation de problèmes spécifiques permettant de formuler des instructions adaptées à un calculateur électronique »⁵.

1 ROBERT (P.), REY-DEBOVE (J.), REY (A.), *Le nouveau petit Robert de la langue française*, Le Robert, 2008, v° « langage » 1.1, p. 1427.

2 *Idem*, v° « langue » 2.4, p. 1428.

3 *Id.*, v° « programmation » 1.2, p. 2038.

4 *Id.*, v° « programme (inform.) » 3.1, p. 2038.

5 *Id.*, v° « langage (inform.) » 1.2, p. 1427.

3. – Divers types de langages de programmation sont apparus au cours de leur histoire, celle-ci commence dans les années 1945-1950 avec des langages extrêmement basiques, et se poursuit toujours avec des langages de plus en plus évolués et de plus en plus spécialisés, de sorte que la conception d'un programme précis s'en trouvera très fortement améliorée. Xavier Linant de Bellefonds enseignait qu'il existe trois types de langages de programmation⁶ : les langages machines, en binaire, les langages assembleurs qui utilisent des systèmes de mnémoniques pour assembler entre eux des morceaux de code binaire, les langages symboliques qui utilisent des signes et symboles en lieu et place de séries de code binaire. Cependant, François Chenique classe les langages symboliques au niveau 1⁷ et précise que les assembleurs sont les logiciels permettant leur traduction en binaire par le biais d'une « table de correspondance »⁸.

Caractérisent ces langages de programmation est : d'abord leur fonction, permettre la programmation ; ensuite leur nature, être un langage, soit « la faculté d'exprimer et communiquer sa pensée par des mots »⁹ ; enfin les éléments qui les constituent, soit un lexique, une syntaxe et une traduction exécutable du code écrit en respectant cet ensemble. Ainsi, selon nous, un langage de programmation est un système d'élaboration d'un programme informatique par le recours à un code constitué d'un lexique, d'une syntaxe et d'une traduction à respecter de façon à permettre directement et automatiquement l'exécution du programme ou la production d'une version exécutable. Alors, le programmeur-utilisateur du langage doit respecter le lexique et la syntaxe. Le concepteur de l'instrument de programmation qui met en œuvre le langage – compilateur, assembleur, interpréteur, environnement d'exécution – doit respecter le jeu des traductions.

4. – La notion de format, qui ne sera qu'accessoirement étudiée, est très proche puisqu'il consiste au respecter d'un mode d'encodage, mais il ne s'agit alors que d'une numérisation ou une transposition numérique d'un élément concevable de manière analogique : une musique, un fichier, une page. La grande différence avec le langage de programmation est que celui-ci s'adresse à un programmeur, tandis que le format de fichier ne s'adresse qu'à la machine par le biais d'un programme qui va automatiquement procéder à la transposition. Il n'y a donc pas d'éléments programmés dans le format de fichier, ni de lexique ou éléments symboliques destinés à la compréhension. Le fichier obtenu, contrairement au programme, ne bénéficie pas en lui-même de la protection du droit d'auteur, la numérisation est indifférente. Certains éléments sont utilisés de

6 HOLLANDE (A.) et LINANT DE BELLEFONDS (X.), *Pratique du droit de l'informatique*, Delmas, 5^e éd., 2002, p. 186, n° 1103.

7 CHENIQUE (F.), *Software . Langages et systèmes d'exploitation*, Dunod, Paris, 1971, p. 29.

8 *Idem*, p. 31.

9 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *CCE* n° 4, avril 2005, étude 15, n° 1, §1.

manière hybride, comme le HTML et le CSS qui permettent la création de pages internet et qui forment une frontière entre les deux notions. Le HTML est un format de fichier lu par un navigateur internet. Il est possible de transposer de manière automatique un document texte en document HTML ; pour autant le HTML n'en reste pas moins un langage de programmation de lignes de code – suites d'instructions – destinées à créer une interface graphique.

5. – La propriété intellectuelle a vocation à protéger la création immatérielle d'un auteur, au sens commun du terme. Il peut s'agir, principalement, d'une invention qui pourra être brevetée en guise de remerciement pour la solution nouvelle, d'un signe distinctif permettant la démarcation, ou d'un apport original au sens du droit d'auteur. Les simples idées sont, quant à elles de libre parcours, encore que certains plaident pour l'application de la théorie civiliste des agissements parasites. Il y a lieu de rechercher à quelle catégorie rattacher les langages de programmation. La Cour de justice de l'union européenne nous a éclairés concernant la qualification juridique des langages de programmation et des formats le 2 mai 2012¹⁰, à première vue en les rejetant du droit spécial des programmes d'ordinateur, mais en les accueillant potentiellement dans le droit commun du droit d'auteur¹¹. Certains s'offusqueront d'une telle décision, d'autres la souligneront, nous chercherons à l'expliquer, car « la question mérite d'être approfondie »¹².

6. – Comment la propriété intellectuelle doit-elle appréhender les langages de programmation ?

7. – L'objet de cette étude est de rechercher en premier lieu si les langages de programmation sont de simples idées, ou ont une valeur technique, économique et surtout une valeur en tant qu'œuvres de l'esprit originales ; valeurs suffisamment considérables pour qu'ils soient protégés (chapitre 1). Cependant, leur protection n'est pas sans poser des difficultés, il faut en tirer les conséquences sur le régime juridique des langages de programmation, notamment leur exploitation qui retentit sur tout l'univers informatique (chapitre 2).

10 CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. c/ world programming Ltd.*

11 SIRINELLI (P.), « Propriété littéraire et artistique », Recueil Dalloz n° 42, 2012, p. 2836 ; SIRINELLI (P.), note sous CJUE, grande Ch., 2 mai 2012, affaire numéro C-406/10, *SAS Institute Inc. contre World Programming Ltd*, *RIDA*, juil. 2012, p. 215.

12 GAUTIER (P.-Y.), *Propriété littéraire et artistique*, Puf, 8^e éd., 2012, n° 130.

Chapitre 1 : La valeur considérable des langages de programmation

8. – D'un point de vue technique et économique un consensus se forme sur la valeur des langages de programmation, ils font l'objet d'importants investissements, car ils sont absolument indispensables dans la conception moderne des logiciels (I). En revanche, leur valeur juridique en qualité d'œuvre de l'esprit originale fait l'objet d'un débat (II).

I) L'opportunité économique et technique favorable à la protection des langages

9. – Grâce à d'importants investissements humains et économiques dans la recherche et le développement de langages de programmation, la conception de logiciels a été particulièrement facilitée. De telles facilités font du langage de programmation une étape technique améliorant la conception de logiciels, une véritable réponse technique à une difficulté d'ordre technique (A), à tel point qu'ils en deviennent indispensables et que l'on puisse douter de l'opportunité de les réserver (B).

A) Le langage : une étape technique améliorant la conception des logiciels

10. – Les langages de programmation constituent des apports techniques indéniables qu'il convient de protéger en tant que tel (1) et en tant qu'investissements et valeurs économiques conséquents (2).

1. Le langage : une solution technique ayant facilité la conception moderne des logiciels

11. – Les langages de programmation évolués sont le tampon entre les algorithmes qui permettent le fonctionnement d'une machine et le programme d'ordinateur qui fait fonctionner cette machine d'une manière bien déterminée. Ces trois éléments sont en réalité indépendants. Un exemple simple, on souhaite procéder au calcul « 2 x 2 » : l'algorithme est la méthode de calcul employée, ici une multiplication ; les symboles (2 et x), leur position et la faisabilité du calcul correspondent au langage utilisé. Le désordre ou l'emploi de symboles incorrects rendent impossible le calcul, de même lorsque le calcul n'est pas envisagé par le créateur du langage. On parle alors du lexique, de la syntaxe et de la sémantique du langage de programmation. Ils sont issus

d'une évolution millénaire¹³, celle des chiffres et de la logique automatique. Ainsi faut-il bien distinguer l'algorithme, qui est la règle mathématique, du langage qui l'emploie, ce qui invalide l'analyse comparative du langage et de la découverte de lois scientifiques¹⁴, la théorie est dans l'algorithme¹⁵.

12. – **Des langages mathématiques fondamentaux aux langages de programmation.** Les mathématiques sont le langage universel et le seul langage fondamental de l'informatique. Tous les autres langages, textuels, n'ont qu'une fonction intermédiaire. Cependant, plusieurs mathématiques sont à l'œuvre. Contrairement aux idées reçues notre système à 10 chiffres, d'origine arabe, n'a rien d'universel et il est même complexe. Il n'a été imposé en France que par le décret napoléonien du 13 brumaire an IX. Le système le plus simple reste celui adopté par les juristes français et les informaticiens : c'est le binaire. Pour autant, en informatique, l'octet a acquis une position majeure.

La mécanique fut à l'origine de l'informatique : on pense avant tout à la pascaline, première calculatrice mécanique inventée au milieu du XVII^e siècle. Puis des mécanismes de plus en plus complexes sont apparus à l'ère de la mécanographie où on ne résonnait qu'en code binaire, en suites de 0 et de 1. Un programme est un processus logique donnant pour réponse oui ou non (1 ou 0). On peut facilement se représenter l'intérêt de ces systèmes qui fonctionnaient avec des cartes perforées : les bonnes réponses à un « QCM » pourraient être gravées sur une de ces cartes et celle d'un candidat sur une autre, le rôle de la machine est de les comparer et sans erreur donner le résultat.

13. – L'ordinateur n'est qu'une version électronique, mais incroyablement plus performante, de cet appareil. Pour des applications plus poussées il faut des programmes plus développés, ce qui exige soit plus d'efforts, soit de trouver des modes de programmation simplifiés. Ainsi l'hexadécimal remplaça rapidement la programmation en binaire, il s'agissait alors de recourir à des suites entières d'instructions binaires par l'emploi de combinaisons de lettres pour accélérer la programmation. Mais y a-t-il véritablement création d'un langage de programmation lorsque l'on se contente d'un processus automatique de traduction ? De l'hexadécimal on est passé à l'octet qui permet de communiquer avec la machine par le biais de l'alphabet occidental et des chiffres arabes. L'octet représente en effet une suite de huit réponses binaires (1 ou 0), par exemple 01100001 correspond à la lettre a¹⁶. Le constat n'est valable que pour les langages de niveau 1 où l'on se

13 Depuis 38 siècles av. J.C. selon SANTOPAULO (M.-A.), *Protection juridique des langages de programmation*, Mémoire DEA, 2002, Montpellier, n° 11.

14 Notamment VIVANT (M.), « 160 - À propos des langages de programmation », in *Lamy Droit du Numérique*, Lamy, 2012, §7 : « On pourrait imaginer que le créateur d'un langage informatique ne puisse s'opposer à son utilisation par des tiers, comme le découvreur d'une loi scientifique, mais soit en droit d'exiger un paiement de qui en userait. ».

15 Cf. cependant *infra*, n° 48.

16 D'après le site : <http://www.convertbinary.com/>

contente de nommer les instructions, alors qu'au niveau 2 on les regroupe en séries pour former des macro-instructions, pour autant la syntaxe reste entièrement dépendante de la machine¹⁷.

Il ne sera véritablement possible de parler de langage de programmation qu'à partir des langages de niveau 3, lorsque la syntaxe s'est développée, en particulier le FORTRAN, édité par IBM dans les années 1950 et destiné au calcul scientifique. Il s'agissait d'offrir au programmeur des briques de travail pour concevoir des logiciels en respectant des règles grammaticales. En effet, celui qui a recours à un langage de programmation suffisamment évolué pourra économiser l'effort du calcul d'un algorithme, en particulier pour la programmation orientée objet. Il lui suffira de suivre les règles de formation du langage. Les briques comme la syntaxe fournissent de l'autocode, surtout concernant la programmation orientée objet. La programmation moderne est un processus presque automatique qui consiste à déterminer la fonctionnalité désirée, c'est-à-dire la sémantique, puis à rechercher dans le langage utilisé les termes qui permettent de la mettre en œuvre, le lexique, tout en respectant la syntaxe.

Cependant, les langages binaires fondamentaux sont toujours utilisés, mais appartiennent davantage au monde du *hardware* que du *software*¹⁸ étant conçus par les producteurs de matériels. En effet, pour fonctionner un processeur doit être commandé. Or, les commandes fondamentales varient en fonction de la machine. Pour l'expliquer, monsieur Higman écrivait que « lorsqu'on la décompose en plusieurs étapes, la phrase "Mange ton repas" commence par "Ouvre ta bouche" si l'on s'adresse à un enfant, mais par "Ouvre ton bec" s'il s'agit d'un perroquet ». La commande « mange ton repas » est écrite dans un langage plus évolué, elle doit se décomposer en plusieurs commandes qui dépendent de la machine sur laquelle elles sont exécutées : « bec » pour un perroquet, « bouche » pour un enfant.

14. – On différencie ainsi le langage binaire, compris par l'ordinateur, du langage évolué compris par l'homme¹⁹. Celui qui crée l'équivalence, la traduction, est le concepteur du langage de programmation. Ce concepteur va écrire chaque élément de lexique et les règles d'utilisation de son langage. Il va ensuite écrire la traduction binaire, ou en langage intermédiaire, de chacune des commandes qu'il est possible d'écrire dans son langage, dans un document voire dans un instrument de programmation (compilateur ou un interpréteur, logiciels qui ont pour rôle de rendre exécutables d'autres programmes). « *Les langages les plus évolués comme les plus élémentaires*

17 CHENIQUE (F.), *Software . Langages et systèmes d'exploitation*, Dunod, Paris, 1971, p. 29-36.

18 « *Hardware* » terme anglais désignant le matériel, par opposition le « *Software* » désigne les logiciels.

19 Dans leur ouvrage technique, monsieur et madame Poyen traitent ces deux notions dans deux chapitres distincts : POYEN (Ja.) et POYEN (Je.), *Le langage électronique*, PUF . Que sais-je ?, 3^e éd., 1967, chapitre III « Traitement du problème . Le langage machine », p. 47-61, et chapitre V « Les langages symboliques », p. 73-105.

consistent dans une équivalence rigoureuse - mais plus ou moins évocatrice à l'esprit de l'opérateur - avec le code binaire, fondement du langage de programmation. »²⁰. Chacun des éléments de ce langage est donc une suite d'instructions binaires et le langage dans son ensemble une collection, une série, de suites d'instructions.

Le concept est simple, à chaque mot du dictionnaire d'un langage correspond en réalité une série d'instructions plus fondamentale que le langage conçu, de sorte que le concepteur s'économise une programmation excessivement longue. Dans les langages de programmation symboliques et évolués « les types d'opérations sont beaucoup plus synthétiques [qu'en langage binaire], chaque opération symbolique pouvant donner naissance à plusieurs opérations machine »²¹. Par exemple, « au mot "feu" lever les fusils, presser la détente, recharger et reposer les armes... feu ! »²². Monsieur Chenique parle à cet égard d'« autocode » ou de « macro-instruction », chaque macro-instruction étant traduite « en une série d'instructions simples ; cette opération s'appelle l'*expansion* de la macro-instruction »²³.

15. – **Invention et langages de programmation.** Monsieur Santopaolo proposait de protéger les langages de programmation par le biais du droit des brevets²⁴. Il part de la protection par un brevet américain d'un format de fichiers, le GIF, et envisage naturellement l'extension aux langages de programmation. Il est vrai que le mode d'encodage dans un format de fichier – ou l'emploi d'un langage de programmation – adapté, permet de gagner de l'espace disque – comme, par exemple, les formats MP3 ou MP4 – ou de rendre l'exécution plus rapide. L'utilisation du matériel est donc plus performante. C'est d'ailleurs le propre d'un langage de programmation d'être plus efficace et plus complet qu'un langage concurrent²⁵. Le fait de recourir à un langage de programmation comme étape dans la rédaction d'un programme relève de la méthode, insusceptible de protection par le droit des brevets²⁶.

En revanche, si l'on considère qu'un langage de programmation est déjà un programme, formalisé au sein d'un compilateur ou interpréteur, et que ce langage déterminé permettait de réaliser sur le matériel un « effet technique supplémentaire », comme l'exige l'Office européen des brevets, alors on devrait en conclure à la brevetabilité d'un tel langage de programmation²⁷. En vérité, le caractère

20 GAUDRAT (P.), « L'invention informatique : un débat difficile et contourné », *RTD Com.* 2005, p. 512, n° 6.

21 POYEN (Ja.) et POYEN (Je.), *Le langage électronique*, PUF . Que sais-je ?, 3^e éd., 1967, p. 73.

22 HIGMAN (B.), traducteur MARGULLICI (M.-T.), *Étude comparative des langages de programmation*, Dunod, 1973, p. 8.

23 CHENIQUE (F.), *Software . Langages et systèmes d'exploitation*, *op. cit.*, p. 32.

24 Il y consacre d'ailleurs la moitié de son étude sur les langages de programmation : SANTOPAULO (M.-A.), *Protection juridique des langages de programmation*, *op. cit.*, n° 119-173.

25 Cf. en ce sens, mais nuançant le propos : *Idem*, n° 36 et 37.

26 L'article L. 611-10 2 a. et c. excluent expressément les méthodes.

27 Ce depuis une décision du CRT de l'OEB 3.5.1, 15 juil. 1986, T208/84, *Vicom*, JO de l'OEB 1987, n° 14.

« supplémentaire » de l'effet technique semble ici indémontrable ; en matière de logiciels, sont en général concernés les codecs, petits programmes permettant l'utilisation du matériel.

C'est sur les logiciels qui mettent en œuvre les langages de programmation, les compilateurs, les interpréteurs voire les environnements d'exécution, que monsieur Santopaolo proposait la protection au titre du droit d'auteur et du droit des brevets. L'affaire *Oracle contre Google*²⁸ a essentiellement portée sur des brevets dont faisait l'objet le langage *JAVA*. Ce n'était pas le langage lui-même qui était protégé, mais le processus utilisé par l'interpréteur qui est le logiciel fondamental mettant en œuvre le *JAVA*²⁹. Même constat pour monsieur Roussel³⁰, la mise en œuvre de méthodes d'analyse d'un langage de programmation dans un instrument de programmation pourrait être brevetée.

16. – À vrai dire, c'est toute la jurisprudence de l'Office européen des brevets sur la brevetabilité des logiciels qui peut être rejetée³¹. Ainsi, envisager la brevetabilité des langages ne semble pas pertinente, sauf à vouloir déformer davantage les conditions de brevetabilité.

Cependant, ce constat ne peut être automatiquement étendu aux langages de programmation fondamentaux, dans la mesure où ils permettent de faire fonctionner le processeur d'une machine de différente façon. Ils devraient donc pouvoir être brevetés, mais alors sous réserve que les fonctionnalités et techniques fondamentales soient nouvelles. Le seuil de la nouveauté et l'apport seront difficiles à apprécier. On peut toutefois énoncer que le tout premier recours au langage binaire en matière informatique, qui est à l'origine de la mécanographie, fut breveté³². Même analyse concernant l'informatique quantique qui ne relève plus du binaire mais d'une logique différente, l'ordinateur quantique tend à faire l'objet de brevets d'invention³³. Il pourrait bien révolutionner la programmation, l'utilisation des processeurs quantiques supposera l'élaboration de nouveaux langages fondamentaux éventuellement brevetables.

17. – Ainsi, le recours au brevet pour protéger les langages de programmation ne sera réservé qu'à certaines situations, toutefois l'apport et la solution technique présupposent un investissement

28 Tranchée par Cour du district Nord de Californie, 23 mai 2012, C10-03561 WHA, *Oracle c/ Google*.

29 BEAUCHAMP-TREMBLAY (Xavier), Droit d'auteur dans les logiciels 101 – Oracle c. Google, ActifsCreatifs [en ligne], 12 déc. 2012, <http://www.actifscreatifs.com/droit-auteur-logiciel-oracle-google/>, consulté le 28 mai 2013.

30 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *CCE* n° 4, avril 2005, étude 15, n° 6, §8.

31 MACREZ (F.), *Créations informatiques : bouleversement des droits de propriété intellectuelle ? . Essai sur la cohérence des droits*, Litec . Collection du CEIPI, 2011, n° 110.

32 Présentant les différentes inventions et brevets entourant le recours à la mécanographie : FAVRE (D.) « L'invention de la mécanographie », Interstices [en ligne], 26 juin 2009, http://interstices.info/jcms/n_51928/linvention-de-la-mecanographie, consulté le 29 mai 2013.

33 Cf. à titre d'exemple le brevet « ordinateur quantique » référencée sur le site Patfr, <http://www.patfr.com/200109/WO0171661.html>, consulté le 10 mars 2013.

et une certaine valeur économique elle-même protégée.

2. *Protection d'investissements et valeurs économiques conséquents*

18. – **L'explosion de la valeur économique des langages.** Simple argument de vente du matériel à l'origine, sous l'influence d'IBM les programmes d'ordinateurs sont peu à peu devenus un marché distinct. Si les supercalculateurs entraînent des coûts de production du matériel nettement supérieur au logiciel, leur part de marché est marginale face au micro-ordinateur. Or, déjà en 1967 le coût de conception des programmes dépassait largement celui du matériel³⁴, lequel restait pourtant considérable même concernant les *mainframes*³⁵. Deux mouvements se dégagent alors : un mouvement scientifique d'universalisation qui prônait l'ouverture des sources, le partage et la standardisation pour qu'un effort commun pousse à l'innovation ; un mouvement industriel dans lequel chaque acteur développait ses propres normes de programmation, donc ses propres langages de programmation pour se démarquer dans les fonctionnalités proposées. La standardisation fut proposée par le projet UNIX, il s'agissait de n'utiliser que les quelques commandes fondamentales communes à tous les processeurs. Il s'agissait ensuite de combiner ces commandes pour en obtenir de plus complexes. Le fonctionnement était alors universalisé, mais conduisait à ignorer les fonctionnalités exceptionnelles offertes de manière directe par certaines machines, sans passer par l'intermédiaire moins efficace des combinaisons. Les constructeurs n'ont donc pas tous admis cette logique. Pendant plusieurs dizaines d'années de conception et dans une logique darwinienne les langages ont évolué en se spécialisant.

19. – Ce sont les algorithmes et les langages de programmation qui ont exigé les investissements les plus lourds. Or, « les investissements dédiés à la création des logiciels paraissent destinés à rester élevés »³⁶, la productivité évoluant lentement. De nos jours les logiciels sont constitués de plusieurs millions d'instructions, le travail d'un concepteur consiste autant à assembler des modules déjà existants qu'à en créer de nouveaux. De plus, économiquement on peut comparer la création de langages de programmation à la création d'une bibliothèque de modules. La « décomposition de la codification en différentes étapes permet de réduire le coût de la codification »³⁷ c'est pour cette raison que les langages de programmation ont évolués vers des formes de plus en plus

34 POYEN (Ja.) et POYEN (Je.), *Le langage électronique*, PUF . Que sais-je ?, 3^e éd., 1967, p. 5.

35 Aussi appelés mini-ordinateurs, ces machines sont exploités collectivement, plusieurs postes se partageant la puissance d'un unique appareil, c'est donc l'inverse de ce qui se fait majoritairement désormais avec la mise en réseau de micro-ordinateurs.

36 MACREZ (F.), *Créations informatiques : bouleversement des droits de propriété intellectuelle ?*, *op. cit.*, p. 50, n° 36.

37 HORN (F.), *L'économie du logiciel*, thèse soutenue le 18 décembre 2000, Lille, accessible sur : <http://www.univ-lille1.fr/bustl-grisemine/pdf/extheses/50374-2000-23-24.pdf>, p. 44.

compréhensible par l'homme, selon monsieur Horn³⁸.

20. – Enfin, en pratique les langages sont mis à disposition par le biais des logiciels qui les mettent en œuvre : les instruments de programmation. Bien souvent l'instrument prend le nom du langage ce qui contribue, malheureusement, à les confondre³⁹. Monsieur Kloetzer parle d'« outils de génie logiciel », il donne à titre d'exemple le prix de *visual studio* : 610 euros hors taxes par mois et par développeur ou 7 200 euros par développeurs pour 2 ans⁴⁰. À ces outils de génie logiciel correspondrait un coût de production et une tarification nettement supérieur aux autres logiciels⁴¹.

Un tel coût a trois justifications : premièrement, il facilite considérablement la programmation⁴², laquelle est une activité plutôt lucrative ; deuxièmement, c'est un programme difficile à concevoir⁴³ ; troisièmement, il suppose en amont de créer un langage de programmation efficace. Si tel n'était pas le cas, la création d'un langage concurrent serait aisée. Or, on ne programme plus autrement que par l'emploi de tels outils, « c'est donc à l'échelle de plusieurs millions d'exemplaires qu'un tel instrument est susceptible d'être reproduit. »⁴⁴.

21. – Ainsi les outils de programmation – le langage et son instrument – et leur mise en service sur différentes machines représenteront des investissements et une valeur économique considérables, incomparables avec un simple programme d'ordinateur.

22. – **La protection de la valeur économique.** Mis à part le droit des brevets qui s'intéresse davantage à l'innovation technique, et le droit d'auteur à l'originalité, la valeur économique des langages peut-être protégée par le biais de divers mécanismes : le droit des marques, le secret et la responsabilité délictuelle.

23. – Comme dans tous les domaines, le droit des marques a vocation à intervenir en matière de langages de programmation. La solution n'a rien d'originale ni même de contestable dès lors que le nom donné au langage est suffisamment distinctif pour être déposé en qualité de marque. Mais le langage lui-même peut-il être un signe distinctif ? Nous savons qu'un langage de programmation est une norme à respecter pour que l'ordinateur puisse interpréter correctement la volonté du programmeur. Des termes précis sont exigés, alors les logiciels d'interprétation ou de compilation

38 *Ibidem*.

39 CHENIQUE (F.), *Software . Langages et systèmes d'exploitation*, *op. cit.*, p. 21.

40 KLOETZER (H.), *Introduction à l'économie numérique*, Lavoisier, 2012, p. 108-110.

41 DRÉAN (G.), *L'industrie informatique . Structure, économie, perspective*, Masson, 1996, p. 223.

42 Ils permettent notamment une automatisation partielle de la conception : HORN (F.), *L'économie du logiciel*, *op. cit.*, p. 46.

43 Le compilateur ou l'interpréteur ne se contentent pas de procéder à une traduction, ils peuvent identifier et corriger des erreurs, de plus la seule compilation suppose de respecter une importante série d'étapes.

44 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *op. cit.*, n° 1, §3.

vont pouvoir les traduire dans un format exécutable par la machine. En matière de page internet, la norme W3C⁴⁵ a été établie. Son respect permet d'assurer la compatibilité des sites internet avec les navigateurs qui vont les lire. On parle d'ailleurs de certification, c'est un système de norme très comparable aux certifications de l'AFNOR tel que la NF, car il s'agit de respecter une syntaxe bien précise. Le taux de respect de la norme W3C, calculable depuis leur site internet, pourra être affiché sur la page. La norme d'un langage de programmation pourrait donc être un signe distinctif de qualité et à ce titre être enregistré en tant que marque. Mais alors est-ce le langage lui-même qui est protégé ou bien la norme ? Pourra-t-on interdire l'exploitation d'une norme divergente ? Le droit des marques ne le permet pas, sauf à démontrer une trop forte proximité, mais un droit d'auteur sur la norme le pourrait. La question sera donc de savoir si une norme est protégeable par le droit d'auteur⁴⁶.

24. – Le secret autour du langage est un intéressant mécanisme dès lors que le développeur veut empêcher la décompilation de son logiciel. En effet, si une société souhaite rendre totalement impossible la décompilation de son logiciel, il lui suffit de développer son propre environnement d'exécution, sa propre machine, son propre langage de programmation puis son logiciel. Ainsi les logiciels conçus à partir du langage secret seraient protégés. Cependant, cela sacrifierait toute possibilité d'interopérabilité avec le matériel d'autres constructeurs et contraindrait la conception de logiciels. C'est, par exemple, l'approche des constructeurs de consoles de jeux vidéos. On notera que le secret doit être conforté par des mécanismes contractuels interdisant sa divulgation tels que les clauses de confidentialités, de secret, voire de non concurrence.

25. – La concurrence déloyale et le parasitisme sont des mécanismes plus convaincants. C'est sur le fondement des articles 1382 et 1383 du Code civil, donc sur la notion de faute de droit commun, que la jurisprudence a consacré ces deux mécanismes. Le fondement est donc bien différent du droit de la propriété intellectuelle, droit réel en principe. En la matière ils vont permettre de condamner l'utilisation par un acteur économique d'un langage créé par un concurrent. Le parasitisme consiste à se placer dans le sillon d'un acteur économique pour bénéficier de ses efforts « sans bourse délier », se conduire comme un parasite, par exemple en réutilisant un langage de programmation créé par un autre.

Le parasitisme peut-il être invoqué pour protéger les langages de programmation ? Écartons d'emblée l'argument qui consiste à réserver l'hypothèse des agissements parasitismes aux relations

45 Normes du *World Wide Web Consortium*, fondé notamment par le créateur du html : Tim Berners-Lee, <http://www.w3.org>, consulté le 27 mai 2013.

46 Cf. *infra*, n° 59-61.

non-concurrentielles. D'abord, parce que la Cour de cassation a rejeté cette analyse⁴⁷, décision de bon sens puisque la situation de concurrence n'empêche pas l'activité parasitaire. Ensuite, parce que les juges recourent au palliatif de la « concurrence déloyale par l'avantage concurrentiel que procurait » le fait de bénéficier des efforts de son concurrent sans bourse délier⁴⁸. Le seul facteur divergeant est alors l'existence d'une relation de concurrence.

26. – Pour autant, le droit commun a-t-il vocation à protéger ce que le droit spécial aurait rendu libre de droit ? Il semble qu'en vertu de la maxime *specialia generalibus derogant*⁴⁹, le parasitisme, et son pendant détourné en concurrence déloyale que nous venons d'exposer, n'ait pas à s'appliquer aux créations immatérielles⁵⁰. « *Peut-on regarder comme fautive par principe l'utilisation d'une création non protégée du seul fait qu'elle permet de profiter des investissements réalisés par un tiers ?* »⁵¹. Les juges du fond l'admettent largement, or, cela représente une atteinte trop importante à la liberté du commerce et de l'industrie selon la Cour de cassation⁵². La volonté du législateur serait de réserver une part de liberté, laisser de libre parcours les éléments non éligible à la protection⁵³. Ne faudrait-il pas interpréter en ce sens les directives européennes portant sur la propriété intellectuelle, notamment le onzième considérant de la directive 2009/24/CE⁵⁴ ? Il entrerait alors en conflit avec l'article 8 de la même directive⁵⁵ qui réserve la possibilité de protéger les créations par des mécanismes contractuels, notamment de secret, ou de la concurrence déloyale. En fait, il n'y a en fait pas conflit mais cohérence, car de tels mécanismes sont en réalité plus exigeants que la propriété intellectuelle, contrairement aux simples agissements parasitaires. « *Le parasitisme parasite-t-il la propriété intellectuelle ?* »⁵⁶. Si cette action était ouverte la réponse

47 Cass. com., 14 fév. 2012, n° 10-27873 : « l'action en responsabilité pour agissements parasitaires peut être intentée par celui qui ne peut se prévaloir d'un droit privatif et ne requiert pas l'existence d'un risque de confusion » .

48 En ce sens : CA Paris, 4e ch., 1^{er} oct. 2008, *Studio 15 Beauté c/ DF France, Éole et Dessange International*.

49 Latin signifiant que la règle spéciale déroge à la règle générale.

50 Appliquant le principe aux créations numériques : MACREZ (F.), *Créations informatiques : bouleversement des propriétés intellectuelles ?*, op. cit., n° 494 et s.

51 LUCAS (A.), LUCAS (H.J.) et LUCAS-SHLOETER (A.), *Traité de la propriété littéraire et artistique*, 4^e édition, 2012, LexisNexis, n° 21.

52 Cass. com., 8 avr. 2008, n° 07-11385.

53 Ceux dont le monopole est expiré ou qui ne répondent pas aux critères de protection.

54 Considérant 11 de la directive 2009/24/CE, 23 avril 2009, concernant la protection juridique des programmes d'ordinateur : « *Pour éviter toute ambiguïté, il convient de préciser que seule l'expression d'un programme d'ordinateur est protégée et que les idées et les principes qui sont à la base des différents éléments d'un programme, y compris ceux qui sont à la base de ses interfaces, ne sont pas protégés par le droit d'auteur en vertu de la présente directive. En accord avec ce principe du droit d'auteur, les idées et principes qui sont à la base de la logique, des algorithmes et des langages de programmation ne sont pas protégés en vertu de la présente directive. Conformément à la législation et à la jurisprudence des États membres ainsi qu'aux conventions internationales sur le droit d'auteur, l'expression de ces idées et principes doit être protégée par le droit d'auteur.* »

55 Art. 8 de la directive 2009/24/CE, op. cit. : « *Les dispositions de la présente directive n'affectent pas les autres dispositions légales concernant notamment les brevets, les marques, la concurrence déloyale, le secret des affaires, la protection des semi-conducteurs ou le droit des contrats.* »

56 HUILIER (J.), « *Le parasitisme parasite-t-il la propriété intellectuelle ?* », *Gazette du Palais*, 7 nov. 2001, p. 6.

serait positive, car il suffit à l'acteur économique de démontrer qu'un tiers utilise ses efforts pour pouvoir le faire sanctionner. Les conditions d'application sont moins strictes.

Ainsi le parasitisme ne devrait jamais être invoqué pour ce qui entre dans le champ d'application de la propriété intellectuelle⁵⁷, en particulier les langages de programmation⁵⁸. D'autant plus qu'ils sont devenus indispensables pour la conception moderne des logiciels.

B) Le langage : une étape technique indispensable à la conception moderne

27. – Certains langages de programmation ont acquis une telle notoriété que les programmeurs ne sauraient passer outre (1), ainsi des arguments fort critiquables sont avancés pour libéraliser les langages de programmation (2).

1. Les langages incontournables

28. – Un langage de programmation n'a d'utilité que si l'utilisateur dispose d'une machine capable d'exécuter les logiciels écrits dans ce langage. Le système d'exploitation est le logiciel au cœur d'un ordinateur, il permet de créer un environnement de travail pour l'utilisateur, il dispose de petits logiciels destinés à faire fonctionner le matériel et les périphériques. Tous les autres logiciels – traitement de texte, jeu-vidéo, simulateur – sont des applications. Ils permettent de faire fonctionner le système, et a fortiori l'appareil, d'une façon différente. Ainsi, pour fonctionner, l'application doit pouvoir communiquer avec le système d'exploitation, il leur faut un langage binaire – ou intermédiaire – commun. Nous l'avons dit⁵⁹, les langages de programmation vont avoir une traduction en binaire, ou intermédiaire, mais il peut y avoir différents langages binaires. Quatre familles de systèmes d'exploitation se partagent les marchés : Windows, Mac, Linux et JAVA⁶⁰. Chacun d'entre eux, en tant qu'univers d'exécution des logiciels d'application, a sa propre façon de gérer les codes sources. Chacun pourra comprendre un ou plusieurs langages binaires – ou langages intermédiaires – dans lequel les langages de programmation devront pouvoir être traduits.

29. – Plus un langage est universel plus il est apprécié, ainsi certains langages sont désormais incontournables et sur eux repose toute la programmation, bien que la volonté de créer un langage de programmation universel soit désormais abandonné⁶¹. Le C – et son dérivé le C++ – est le

57 Peu importe qu'il soit effectivement protégé ou non, dès lors qu'il s'agisse d'une création immatérielle.

58 En ce sens : MACREZ (F.), « Le droit d'auteur, le programme d'ordinateur et la cour de justice », *RIDA*, 22 octobre 2012, §final ; dans le sens contraire : ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *op. cit.*, n° 7, *in fine*.

59 *Cf. supra*, n° 14.

60 En tant que machine virtuelle.

61 Monsieur et madame Poyen sont revenus sur leur propos en 1969, ils affirmaient en 1960 l'émergence d'un

langage de programmation le plus enseigné, il permet de créer des programmes compatibles avec les trois principaux systèmes d'exploitation.

Le HTML et le CSS sont deux langages de programmation qui permettent de créer des pages internet. Ils sont ainsi à mi-chemin entre le langage de programmation et le format de fichier. De nombreux logiciels permettent d'écrire de manière automatique une page internet à partir du modèle d'une page de texte standard. Contrairement à la plupart des logiciels le code source d'une page internet est directement exécuté par la machine par le biais du navigateur. Ce langage a évolué au grès des conventions (de l'HTML 1 à l'HTML 5), il connaît certaines variantes non reconnues par le W3C⁶² et en principe invalides. Ce langage est tant répandu et utilisé que les nombreuses erreurs dans les codes amateurs conduisirent les concepteurs de logiciels de navigation à prévoir des interprétations pour ces erreurs de programmation.

Le JAVA a sa logique propre, il permet de créer véritablement des logiciels qui seront indirectement exécutables sur toutes les machines en passant par le biais d'une machine virtuelle. Un logiciel, la machine virtuelle de JAVA, va se charger d'exécuter les programmes écrits en JAVA et ensuite transmettre les instructions au système d'exploitation. Il s'agit donc d'un précieux outil qui a fait l'objet d'un contentieux important, en témoigne les affaires *Sun Microsystem contre Microsoft*⁶³ et *Oracle contre Google*⁶⁴.

30. – Il n'est pas rare que les éditeurs de systèmes d'exploitation prévoient des langages propriétaires dont ils imposent l'utilisation pour pouvoir écrire des applications exécutables sur leur système. Le plus important n'est pas d'avoir accès à ce langage de programmation, mais bien aux traductions binaires exécutables par le système. On pourrait éventuellement parler d'interface. Le langage de programmation est donc une version dérivée du langage binaire, il permet de l'exploiter à travers le logiciel de traduction qu'est le compilateur. Il n'est pas aisé de créer un langage concurrent dans la mesure où on recherche un résultat binaire très précis. La règle selon laquelle pour bien programmer il faut un langage spécialisé est toujours valable. La difficulté est la même concernant les formats de fichiers dits propriétaires, ils supposent de recourir à des logiciels particuliers pour pouvoir lire des fichiers. Les clefs binaires – ou en langage intermédiaire – d'interopérabilité – ou de décryptage des formats de fichier – sont des obstacles à la liberté du commerce, cependant ils permettent de contrôler l'usage fait des fichiers.

langage universel, probablement l'Algol : POYEN (Ja.) et POYEN (Je.), « Langage et automatisation » *in Points de vue sur le langage*, Éd. Klincksieck, Paris, 1969.

62 *World Wide Web Consortium*, cf. *supra*, n° 23.

63 Cour du district de San Jose, 9 sept. 1998, *Sun contre Microsoft*.

64 Tranchée par Cour du district Nord de Californie, 23 mai 2012, C10-03561 WHA, *Oracle c/ Google*.

31. – Le rapprochement avec l’AFNOR est intéressant, cette marque de certification de qualité est issue du droit de la consommation, tout comme doit l’être la réglementation des mesures techniques de protection et les langages propriétaires. Réserver l’exécution de logiciels ou la lecture de fichiers à certains acteurs conduit inévitablement à une logique corporatiste et de monopole. L’actuelle exception de décompilation à des fins d’interopérabilité nous semble insuffisante puisqu’elle se contente d’autoriser la décompilation et non d’autoriser l’interopérabilité elle-même par reproduction des éléments logiciels la permettant⁶⁵. La réservation des langages de programmation est donc critiquable.

2. *Les arguments libéraux opposés à la réservation des langages et des formats*

32. – On ne reprochera pas aux mouvements dits "libres" de s’opposer à l’appropriation des langages de programmation, notamment par le biais du droit d’auteur. Leurs revendications, purement politiques ou idéologiques, portent sur l’ensemble de la propriété intellectuelle. En revanche, il nous paraît peu pertinent, d’un point de vue juridique, de rejeter la protection des langages de programmation du seul fait de leur caractère utile⁶⁶, bien que leur privatisation soulève des difficultés d’interopérabilité.

33. – **Le langage : une ressource essentielle.** Les langages de programmation sont devenus des « passages obligés »⁶⁷, pour Michel Vivant, les privatiser conduit à « restreindre à l’excès le domaine public » et « empêcherait toute activité de développement logiciel »⁶⁸. À vrai dire, c’est le cas de toute la propriété intellectuelle ou de toute propriété privée. Pour Christophe Caron leur réservation serait « socialement répréhensible »⁶⁹, tandis que pour Xavier Linant de Bellefonds il a vocation à tomber rapidement dans le domaine public, mais n’y est pas par nature⁷⁰. Un langage de programmation est un outil, comme le pinceau du peintre, ce n’est pas parce qu’un outil est indispensable qu’il ne puisse faire l’objet d’un droit de propriété. Patrick Roussel, expert en

65 Cf. cependant sur ce point : *infra*, n° 118-122.

66 Comme le prétend BOT (Y.), *Conclusions de l’Avocat général*, CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. contre world programming Ltd* BOT (Y.), *Conclusions de l’Avocat général*, CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. contre world programming Ltd.*, cf. *infra*, n° 46-50.

67 Utilisant cette formule : LE STANC (C.), « Droit de l’informatique », *La semaine juridique entreprise et affaires* n° 21, 23 mai 1996, 559, n° 6, §6 ; VIVANT (M.), « 160 - À propos des langages de programmation », *op. cit.* §2. Reprenant l’idée en citant Cass. civ. 1^{re}, 17 juin 2003, n° 01-17,659 ; ROUSSEL (P.), « La maîtrise d’un langage de programmation s’acquiert par la pratique », *op. cit.*, n° 3, §3.

68 VIVANT (M.), « 160 - À propos des langages de programmation », *op. cit.* §2 ; cf. aussi en ce sens LINANT DE BELLEFONDS (X.), « Les outils logistiques du net et la propriété intellectuelle », in *Internet saisi par le droit*, édition des Parques, 1997, p. 147.

69 CARON (C.), *Droit d’auteur et droits voisins*, Litec . LexisNexis, 2^e éd., 2009, n° 178, §2.

70 LINANT DE BELLEFONDS (X.), *Droits d’auteur et droits voisins*, Dalloz . Cours, 2^e éd., 2004, n° 261 (Les éléments non protégeables), §4 ; HOLLANDE (A.) et LINANT DE BELLEFONDS (X.), *Pratique du droit de l’informatique*, *op. cit.*, p. 193, n° 1124.

informatique près la cour d'appel de Montpellier, fait d'ailleurs un important rapprochement entre langage de programmation et système d'exploitation⁷¹. Du fait que les systèmes d'exploitation sont indispensables devraient-ils être libres ? Le seul à l'être est bien Linux⁷², or sa part de marché sur les ordinateurs personnels est faible, c'est donc que le consommateur accepte volontier de payer pour ce qu'il pense être de meilleure qualité. Il en est de même pour les logiciels d'application⁷³ et les langages n'échappent pas à cette logique. Ce qui caractérise un bon langage est son efficacité et son universalité⁷⁴. Si un important développement permet un langage bien plus efficace, pourquoi son concepteur ne pourrait-il en tirer profit lorsqu'il fait bénéficier de cette évolution à des concepteurs de logiciels prêts à en payer le prix ?⁷⁵ Ainsi la protection des langages de programmation est bénéfique en ce qu'elle permet l'apparition de langages améliorés. Elle favorise l'innovation par la recherche d'un monopole⁷⁶, lequel doit être légalement maintenu pour permettre le retour sur investissement. La loi de Moore extrapolée au *software*⁷⁷ enseigne que le langage sera rapidement désuet, nécessitant son renouvellement perpétuel ou l'apparition de nouveaux langages. Au contraire la libéralisation tend à la diversification, ce qui peut poser des problèmes d'interopérabilité.

Il reste l'hypothèse des langages propriétaires indispensables sur certains systèmes. En vertu de la théorie des facilités essentielles, laquelle est appliquée à la propriété intellectuelle depuis la jurisprudence *Magill*⁷⁸ et même appliquée à la propriété littéraire et artistique dans l'affaire *Deezer contre Universal*⁷⁹, si une ressource est absolument indispensable pour exercer sur un marché dérivé, le propriétaire a l'obligation de consentir une licence à un prix raisonnable⁸⁰. C'est donc le droit de la concurrence qui restreindra le monopole sur le langage⁸¹, ainsi la privatisation des

71 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *op. cit.*, n° 1 ; *cf.* aussi en ce sens SANTOPAULO (M.-A.), *op. cit.*, n° 19.

72 Parmi les trois systèmes d'exploitation les plus utilisés : windows, linux, mac-os.

73 Microsoft office est la suite bureautique la plus distribuée alors qu'il existe de nombreux logiciels libres : libre, open, abiword, etc.

74 *Cf.* en ce sens, mais nuanciant le propos : SANTOPAULO (M.-A.), *Protection juridique des langages de programmation*, *op. cit.*, n° 36 et 37.

75 BERTAND (A.), « Chapitre 202 Logiciels » in *Droit d'auteur*, éd. Dalloz, 2013, n° 202.53, §5 : « Le droit d'auteur étant un droit absolu, il appartient à l'auteur qui met au point un nouveau langage informatique de décider s'il va le mettre dans le domaine public pour faciliter le développement de l'informatique. Mais si ce langage présente des qualités (compacité, facilité de programmation...), il est également logique qu'il s'en réserve l'exploitation. »

76 *Cf.* la théorie de l'innovation de SHUMPETER (J.), *Capitalisme, socialisme et démocratie*, Petite Bibliothèque Payot, 1974.

77 Le *hardware*, anglais signifiant le matériel, connaît une évolution significative tous les 6 mois, l'opposé *software* désigne l'informatique immatérielle, principalement les logiciels, l'évolution est moins importante mais notable.

78 CJCE, 6 avril 1995, RTE et ITP Ltd c/ Commission, décision 241/91 et 242/91.

79 TGI Paris, ord. réf., 5 sept. 2011, SAS Universal music France c/ SAS Blogmusik.

80 Cependant la théorie ne permet d'accorder que l'essentiel au demandeur selon : Cass. com., 12 juil. 2005, n° 04-12.388, *NMPP*.

81 VIVANT (M.), *À propos des langages de programmation*, in *Lamy droit du numérique*, Lamy . Droit de l'immatériel, n° 160 ; LAMZOUHRI (A.), *L'incidence de la protection du droit d'auteur sur le logiciel*, mémoire,

langages de programmation n'est pas à craindre.

34. – **Le besoin d'interopérabilité.** Les divergences de formats et de langages de programmation posent des problèmes de compatibilité qui peuvent fausser la concurrence et nuire à l'égalité entre les consommateurs⁸². L'histoire des langues nous montre que le vainqueur est souvent l'acteur le plus puissant : les latinophones au début de notre ère, les francophones au début du siècle dernier, les anglophones depuis un siècle. À plusieurs reprises Microsoft a été condamné du fait de sa situation de monopole⁸³, à l'instar d'IBM dont l'influence n'est plus à démontrer. Le standard ne doit pas être impérativement recherché au risque d'entraîner la stérilité de l'informatique⁸⁴. Le modèle plus ou moins libre d'Unix a entraîné un mouvement inverse à la standardisation, contrairement à la volonté initiale, chaque acteur économique voulant créer sa propre variante⁸⁵. Mais alors la difficulté dépasse les seuls formats et langages de programmation pour atteindre l'intégralité de la propriété intellectuelle, c'est le problème des mesures techniques de protection en général. Il serait souhaitable qu'une norme et des équivalences de traduction dans les langages de programmation se dégagent. La pratique semble imposer une standardisation des mesures de protection ; ainsi que la concession de licences à des conditions « raisonnables et non discriminatoires »⁸⁶, lorsque des éléments protégés sont reproduits. La pratique s'étendra peut-être aux langages. Pour monsieur Chen, l'apparition de « standards ouverts »⁸⁷ serait l'idéal, mais les intérêts divergents des acteurs du marché ne le permettent pas⁸⁸.

35. – Pour monsieur Bertran, dès lors qu'une création devient un standard soit du fait de la pratique soit du fait de l'adoption en tant que marque de certification majoritaire, alors elle tombe dans le domaine public. Il adopte une telle réflexion à propos des interfaces physiques⁸⁹. Le langage

21 mai 2012, Paris-Ouest, accessible sur : <http://www.e-juristes.org/wp-content/uploads/2012/05/Lincidence-de-la-protection-du-droit-dauteur-sur-le-logiciel.pdf>.

82 LUCAS (A.), LUCAS (H.J.) et LUCAS-SHLOETER (A.), *op. cit.*, n° 988.

83 Condamnation à donner accès aux informations permettant l'interopérabilité de *Windows* avec d'autres systèmes : TPICE, 17 sept. 2007, T-201/04, Microsoft c/ Comm. CE, JO UE n° C279, 10 nov. 2007, p. 45.

84 DRÉAN (G.), *L'industrie informatique. Structure, économie, perspective*, Masson, 1996, p. 238.

85 *Idem*, p. 245.

86 BREESE (P.) et KERMADEC (Y.), *La propriété intellectuelle au service de l'innovation*, Nathan. INPI. Repères pratiques, Paris, juil. 2009, p. 155 *in fine*.

87 Définis par l'art. 4 de la loi pour la confiance en l'économie numérique du 21 juin 2004 comme étant « tout protocole de communication, d'interconnexion ou d'échange et tout format de données interopérable et dont les spécifications techniques sont publiques et sans restriction d'accès ni de mise en oeuvre ».

88 CHEN (S.-T.), *Droit d'auteur et protection technique des œuvres – étude de droit comparé européen et asiatique*, thèse soutenue le 13 déc. 2010, Nantes, p. 315, n° 495.

89 BERTAND (A.), « Chapitre 202 Logiciels » in *Droit d'auteur*, éd. Dalloz, 2010, n° 202.26 : « La plupart des interfaces physiques sont dans le domaine public : soit parce qu'elles répondent à des normes adoptées par l'Association française de normalisation (Afnor), qui peuvent de surcroît être rendues obligatoires aux industriels (D. no 84-74, 26 janv. 1984, JO 1er févr., p. 490.) ; soit parce qu'elles constituent un « standard », c'est-à-dire qu'elles correspondent à des spécifications adoptées volontairement par de nombreuses sociétés de la branche professionnelle concernée ».

de programmation universel n'existe pas, mais force est de constater que les langages de programmation se concurrencent rarement. Ils ne sont en général évincés que par des langages plus évolués ou plus spécialisés. Les différents projets d'unification n'ont pas abouti, les langages étant des œuvres profondément évolutives.

En vérité la réservation de l'œuvre ne sera pas en elle-même un obstacle à l'interopérabilité, il s'agit simplement d'aménager le droit d'auteur afin d'éviter que l'utilisateur ne soit captif d'un langage de programmation⁹⁰. Ce fut une solution dégagée dans l'arrêt *SAS contre WPL* rendu par la Cour de justice de l'union européenne le 2 mai 2012⁹¹ et qui sera développée en fin d'étude⁹². Le véritable obstacle à la réservation serait la disqualification des éléments du langage en simples idées.

II) La protection principale des langages de programmation par le droit d'auteur

36. – C'est le plus grand principe du droit d'auteur qui en fixe positivement les critères d'application : « les idées sont de libre parcours, leur formalisation, dès lors qu'elle porte l'empreinte de la personnalité de leur auteur sont protégées au titre du droit d'auteur »⁹³. Négativement, sont indifférents : le mérite, la destination, l'absence de formalisme, le genre et la forme d'expression, l'irrespect de l'ordre public⁹⁴. C'est surtout la condition de forme qu'il nous intéressera d'étudier, la condition d'originalité, d'appréciation souveraine des juges du fond, étant essentiellement renvoyée à des développements ultérieurs⁹⁵.

Le mouvement, confirmé par l'arrêt *Cryo* de juin 2009⁹⁶ consiste à analyser de manière éclatée la qualification juridique des créations numériques. Depuis l'arrêt *Infopaq*⁹⁷, il faudrait d'ailleurs apprécier une contrefaçon en fonction des éléments de l'œuvre dans lesquels réside précisément l'originalité⁹⁸. Ainsi, en surface, un langage n'est qu'une théorie de la programmation agrémentée d'éléments éventuellement protégés par droit d'auteur, mais n'est pas en lui-même protégeable. Cette position défendue par une doctrine importante est faussée (A), car un langage est en réalité une programmation fondamentale dont l'utilisation est réglementée (B).

90 La directive 91/250/CEE, *op. cit.*, prévoit d'ailleurs des exceptions au droit d'auteur sur les programmes.

91 Opérant une distinction entre contrefaçon du logiciel et contrefaçon du langage et permettant la création d'un logiciel similaire par le biais d'exceptions au droit d'auteur : CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. c/ world programming Ltd.*

92 Cf. *infra*, n° 117-127.

93 DESBOIS (HENRI), *Le droit d'auteur en France*, Dalloz, 3^e éd., 1978.

94 Art. L. 112-1 du Code de la propriété intellectuelle.

95 Cf. *infra* n° 81-83.

96 Cass. civ. 1^{re}, 25 juin 2009, n° 07-20.387, *Cryo*.

97 CJUE, 16 juillet 2009, C-5/08, *Infopaq International* ; CCE 2009, comm. 97, note CARON (C.) ; JCP G, 2009, 272, note MARION (L.) ; Propr. Intell., 2009, n° 33, p. 379, note BENABOU (V.-L.).

98 Selon : MACREZ (F.), « Le droit d'auteur, le programme d'ordinateur et la cour de justice », *RIDA*, 22 octobre 2012, note de bas de page n° 32.

A) La vision de surface des langages de programmation : une règle abstraite agrémentée d'éléments protégés

37. – Dans une analyse de surface, l'originalité d'un langage de programmation ne semble envisageable que dans ses agréments : les termes employés, la bibliothèque, le titre du langage (1). Le langage en lui-même semble alors n'être qu'une théorie de la programmation, ce qu'infirmes la jurisprudence (2).

1. Le langage : une œuvre agrémentée d'éléments originaux

38. – Le langage est constitué de mots alors que le langage créé est constitué de néologismes potentiellement originaux, ou de termes du langage commun recyclés. L'ensemble de ces mots est compilé dans un dictionnaire, on parlera alors d'une base de données. Il n'est pas rare que le langage soit nommé, or le titre original d'une œuvre est protégé et peut aussi être déposé en tant que signe distinctif.

39. – **Plusieurs fondements pour la protection du lexique.** Selon l'arrêt *Griolet*⁹⁹ on ne saurait protéger le vocabulaire en tant que tel. Deux raisons peuvent être mises en avant, d'abord même désuet il est depuis fort longtemps dans le domaine public, ensuite toute œuvre ayant « valeur normative » sera exclue de la protection du droit d'auteur¹⁰⁰. Il en est ainsi des textes législatifs et réglementaires officiels, il doit en être ainsi de tous les termes admis par l'académie française, leur orthographe et leur sens étant fixés par elle.

Pourtant, nul doute qu'un néologisme puisse être protégé, du fait de l'indifférence du mérite¹⁰¹ une œuvre littéraire peut prétendre à la protection quelle que soit sa taille. Tout comme pour l'écriture du code source d'un logiciel, le choix du créateur dépend majoritairement de la mise en œuvre d'une logique automatique ou bien de l'arbitraire pur et simple. Or, le droit d'auteur n'appréhende pas les choix logiques ni l'arbitraire, mais le « choix arbitraire original »¹⁰². Bien souvent, le langage de programmation utilisera de simples lettres ou abréviations rendant encore plus difficile la recherche de l'originalité. Si l'on prend l'exemple du HTML, la balise « <p> » permet d'ouvrir un paragraphe, la balise « » permet d'insérer une image. Mais il peut arriver que le nom

99 CA Paris, 14 janv. 1992, *Griolet c/ Vautrin*.

100 LUCAS (A.), LUCAS (H.J.) et LUCAS-SHLOETER (A.), *op. cit.*, n° 97.

101 Art. L. 112-1 du Code de la propriété intellectuelle.

102 CARON (C.), *Droit d'auteur et droits voisins*, Litec . LexisNexis, 2e éd., 2009, n° 94.

d'une balise soit légèrement décalée par rapport à sa fonction, c'est le cas de « » qui permet de mettre en évidence un mot dans un paragraphe, par défaut en le mettant en caractères gras.

D'éventuels néologismes dépassant les simples abréviations pourront apparaître, mais seront rares¹⁰³. Pour autant, certains concepteurs font preuve d'une imagination débordante, leur langage de programmation pouvant ressembler à des recettes de cuisine, au sens propre évidemment, ou au langage des vaches¹⁰⁴.

40. – Les termes d'un langage de programmation seraient-ils des titres d'œuvres ? On peut, du moins, s'inspirer de ce régime pour justifier la protection des néologismes. Ils bénéficieraient alors d'un critère d'originalité objectivé¹⁰⁵ et d'une protection spéciale contre l'emploi du terme par un concurrent. Si l'on considère qu'un mot dans un langage de programmation est en fait un raccourci vers un morceau de code programmé en binaire auquel il renvoie¹⁰⁶, ne peut-on considérer qu'il s'agisse du titre de ce mini-programme ? L'intérêt de l'approche est discutable, il s'agirait de bénéficier du régime dérogatoire des titres d'œuvres, mais là encore il n'est pas certain qu'un régime spécial leur soit attribué. Desbois enseignait que l'article L. 112-4 du Code de la propriété intellectuelle est un hybride entre droit des marques, droit d'auteur et droit de la concurrence¹⁰⁷. Il semblait aussi distinguer deux degrés d'originalité en la matière : l'originalité absolue¹⁰⁸ du néologisme, ou groupe de mots, qui permettrait d'accéder à la protection du droit commun du droit d'auteur ; l'originalité simplement relative¹⁰⁹, qui permettrait de le protéger en tant que titre dans sa relation avec l'œuvre qu'il désigne, par le biais du régime subsidiaire de l'article L. 112-4. Cette hypothèse d'un double degré d'originalité des titres justifierait que les titres relativement originaux soient protégés « en contemplation de l'œuvre »¹¹⁰ et que des atteintes prétoriennes leur soient portées¹¹¹. La balise « » du HTML, « fort » en anglais, si elle ne peut être absolument originale peut être relativement original par rapport à l'œuvre titrée, laquelle correspond à une suite

103 Pour une étude comparée de différents langages de programmation de haut niveau présentant de nombreux termes en langages de programmation : BELLEC (M.) et LORET (P.), *L'outil informatique, pour les sciences et les sciences économiques. Des langages aux applications*, Masson, Paris, 1987.

104 Cf. à ce propos une page internet dédiée à ces langages de programmation « exotiques, ésotériques, délirants, hilarants » : HUNAULT (G), [petite] *histoire des langages de programmation* [en ligne], <http://www.info.univ-angers.fr/~gh/hilapr/detour.htm>, consulté le 28 mai 2013.

105 LUCAS (A.), LUCAS (H.J.) et LUCAS-SHLOETER (A.), *op. cit.*, n° 114, §4.

106 Pour les aspects techniques cf. *supra* n° 14, pour les conséquences juridiques cf. *infra*, n° 52-54.

107 DESBOIS (H.), note sous Trib. Comm. Seine, 26 juin 1951, *Les hauts de Hurlevent*, RTD Com. 1952, p. 763, §10.

108 Envisageant l'existence de « titres nus » créé par un « barreur de titres » en attente d'une œuvre à servir : *idem*, §7-8.

109 *Id.*, §7 : « Sa mission est de servir de pavillon à une œuvre de l'esprit, comme la marque à un produit de l'industrie : il importe de le protéger, non en lui-même, mais comme accessoire de l'œuvre. ».

110 BRUGUIÈRE (J.-M.) et VIVANT (M.), *Droit d'auteur et droits voisins*, Précis Dalloz, 2012, n° 152.

111 LUCAS (A.), LUCAS (H.J.) et LUCAS-SHLOETER (A.), *op. cit.*, n° 114, §3.

d'instructions – en binaire ou en langage intermédiaire – permettant de mettre en évidence du texte sur une page web. Mais ce double degré d'originalité n'est-il pas condamné d'avance ?¹¹² La possibilité de recourir à la concurrence déloyale et aux agissements parasitaires est formellement consacrée, la maxime *specialia generalibus derogante* est cette fois favorable à l'emploi de ces mécanismes¹¹³, car « nul ne peut, même si l'œuvre n'est plus protégée [...], utiliser ce titre pour individualiser une œuvre du même genre, dans des conditions susceptibles de provoquer une confusion »¹¹⁴. L'alinéa prohiberait donc l'utilisation d'un élément lexical d'un langage qui serait original par rapport à la suite d'instructions qu'il désigne, sa fonction. À défaut d'originalité, la concurrence déloyale notamment par agissements parasitaires sera sanctionnée proportionnellement à l'emprunt fait au langage de programmation imité ou copié.

41. – Un arrêt récent est venu perturber l'analyse classique et confirmer celle que nous venons de développer. La Cour d'appel de Paris¹¹⁵ a jugé que le titre *Karbone 14* donné à un jeu ne pouvait être protégé par le droit d'auteur puisque le jeu lui-même n'était pas original. Rappelons que l'alinéa premier de l'article L. 112-4 du Code de la propriété intellectuelle énonce que le titre d'une œuvre est protégé « comme l'œuvre elle-même », ainsi que le relève le professeur Brugière¹¹⁶. Dans l'absolu, une suite très courte de mots devrait pouvoir être protégée, peu importe qu'elle soit utilisée ou non en guise de titre. En l'espèce ce ne fut pas le cas, inutile de s'y attarder¹¹⁷. La Cour s'est en fait prononcée sur le droit d'auteur spécial applicable au titre d'une œuvre, en relevant comme condition à la protection par le droit d'auteur d'abord que le titre désigne une œuvre de l'esprit original, ensuite qu'il soit lui-même gratifié d'une certaine originalité. N'y-a-t-il pas ici application de la maxime *accessorium sequitur principale* ? La question sera étudiée¹¹⁸.

42. – **Une base de données, un titre et un descriptif protégeable.** Comme le soulève Marie-Eugénie Laporte-Legeais à propos des outils de traduction des linguistes¹¹⁹ : le dictionnaire du langage de programmation ou glossaire de la langue est lui-même protégé par le droit d'auteur en tant que base de données en vertu de l'article L. 112-3 du Code de la propriété intellectuelle. Encore faut-il qu'il y ait originalité dans le choix ou la disposition des matières. Mettons de côté le droit *sui*

112 Cf. *infra*, n° 81-83.

113 Contrairement à nos observations quant au droit commun du droit d'auteur, cf. *supra* n° 25.

114 L. 112-4 Alinéa 2 du Code de la propriété intellectuelle.

115 CA Paris p. 6, 1^{re} ch., 17 oct. 2012.

116 BRUGIÈRE (J.-M.), « Protection du titre d'un jeu (non) » in « Droit d'auteur et droits voisins » (chronique), Propriétés intellectuelles n° 46, janv. 2013, p. 45-46.

117 Comme le relève BRUGIÈRE (J.-M.), *ibidem*, protéger en soi la suite de mots *Karbon 14* n'a pas de sens.

118 Latin signifiant : « l'accessoire suit le principal », cf. *infra*, n° 86-88.

119 LAPORTE-LEGEAIS (M.-E.), « Traductions juridiques et droits intellectuels » in *Traduction du droit et droit de la traduction*, sous la dir. de CORNU (M.) et MOREAU (M.), Dalloz, 2011, p. 278.

generis des bases de données qui permet d'interdire toute extraction substantielle dès lors qu'il y a eu un effort substantiel dans la constitution de la base, car les conditions d'application ne seront pas remplies s'il n'y a d'effort que dans la création des éléments et non dans leur référencement et compilation¹²⁰. Nous le verrons, un langage de programmation est aussi une compilation de suites d'instructions, de « mini-programmes »¹²¹, à laquelle le régime de l'article L. 112-3 s'applique.

43. – Si les titres de logiciels sont souvent peu originaux, ceux des langages le sont encore moins. Il s'agit la plupart du temps d'un acronyme renvoyant à un titre purement descriptif. Ils ne pourront en général aspirer qu'au premier degré d'originalité, l'originalité objective, de l'article L. 112-4 du Code de la propriété intellectuelle. C'est le cas du Fortran, du Cobol, du Cobra ou d'Euphoria dont les titres révèlent un souci d'esthétisme linguistique ou un décalage avec l'œuvre titrée, c'est-à-dire une originalité objective. Tout comme pour les termes utilisés dans le langage de programmation, l'originalité absolue sera rarement atteinte. Nous l'avons vu¹²², la prudence commande de recourir au droit des marques pour se prémunir contre les aléas du droit d'auteur¹²³ ainsi qu'au droit de la concurrence déloyale, expressément visé à l'article L. 112-4¹²⁴.

44. – Le descriptif du langage est protégé comme toute œuvre à caractère scientifique ou technique, peu importe que le sujet soit ou non protégeable. Cette protection est strictement indépendante de notre étude et se rapproche davantage des manuels d'utilisation, lesquels sont couverts par le droit commun du droit d'auteur¹²⁵. Typiquement, le monopole sur le langage ne permettrait d'interdire l'élaboration d'un manuel d'utilisation que si celui-ci reproduisait les éléments originaux du langage.

45. – Tous ces éléments incorporés dans un langage peuvent aspirer à la protection du droit d'auteur comme susceptible d'être des formes originales. Toutefois ce n'est pas le langage lui-même qui est alors protégé par le droit d'auteur, mais une base de données, son titre et ses contenus originaux. Or, la Cour de justice est très claire, les langages peuvent « bénéficier, en tant qu'œuvres, de la protection par le droit d'auteur, en vertu de la directive 2001/29, s'ils sont une création intellectuelle propre à leur auteur »¹²⁶. D'ailleurs, il ne serait pas pertinent de viser expressément les langages de programmation dans la directive de 1991¹²⁷ s'ils ne présentaient aucune spécificité par

120 CJUE, 9 nov. 2004, C-203/02, *The British Horseracing Board Ltd e.a. c/ William Hill Organization Ltd*.

121 *Cf. infra*, n° 52-54.

122 *Cf. supra*, n° 23.

123 CARON (C.), *Droit d'auteur et droits voisins*, Litec . LexisNexis, 2^e éd., 2009, n° 132.

124 *Cf. supra*, n° 40-41.

125 CJUE, 2 mai 2012, n° C-406/10, *SAS Institute Inc. c/ world programming Ltd*, point 70.

126 *Idem*, points 44-46.

127 Onzième considérant de la directive 91/250/CEE, 14 mai 1991, concernant la protection juridique des programmes d'ordinateur.

rapport à de simples néologismes et leur compilation. La spécificité des langages de programmation par rapport aux autres œuvres réside dans les règles qui conditionnent la programmation, c'est à tort que cette spécificité conduit à analyser les langages comme de simples théories de la programmation.

2. *Le langage : plus qu'une simple théorie de la programmation*

46. – Mis à part les quelques éléments accessoires au langage de programmation, celui-ci est-il « une œuvre à part entière »¹²⁸ ou le « mode d'emploi d'un ordinateur »¹²⁹, la doctrine ne tranche pas unanimement la question ni au regard de textes trop flous, ni au regard d'une jurisprudence très subtile.

47. – **Des textes interprétables à souhait.** Selon le considérant 14 de la directive logiciel de 1991¹³⁰ : « les idées et principes qui sont à la base de la logique, des algorithmes et des langages de programmation ne sont pas protégés en vertu de la présente directive ». Ce n'est que l'application du principe de non appropriation des idées¹³¹ ; ce qui signifie, a contrario, que la formalisation des langages de programmation doit être protégée par le droit d'auteur¹³². Le défaut d'une telle analyse est qu'elle puisse être étendue aux algorithmes, strictement rejetés de toute protection¹³³. À titre d'exemple, la solution japonaise est beaucoup plus explicite dans sa non-protection des langages de programmation depuis 1985, par ailleurs, la valeur d'un simple considérant est à relativiser. Même dans l'hypothèse où le considérant 14 rejeterait la protection des langages de programmation, sa portée se limite au droit applicable aux logiciels, son article 9.1¹³⁴ permettant à la directive DADVSI¹³⁵ de protéger les langages de programmation. Enfin, un langage de programmation incorporé dans un programme ne serait-il pas un mode d'expression de ce programme ? Ou bien le langage de programmation ne serait-il pas une étape de conception préparatoire ? À l'un ou à l'autre

128 LINANT DE BELLEFONDS (X.), *Les outils logistiques du net et la propriété intellectuelle*, in *Internet saisi par le droit*, éd. des Parques, 1997, p. 147 et 149

129 GRELON (B.), *Les entreprises de services*, Economica, 1978, n° 315

130 Et le considérant 11 de la directive 2009/24/CE 23 avril 2009, concernant la protection juridique des programmes d'ordinateur.

131 La majorité de la doctrine se prononce en ce sens, même celle qui rejette purement et simplement toute protection aux langages de programmation, cf. notamment : DERCLAYE (E.) et STROWEL (A.), « Droit d'auteur et numérique : logiciels, bases de données, multimédia », Bruylant, Bruxelles, 2001, p. 192, n° 218.

132 Analyse énoncée mais rejetée par : POLLAUD-DULLIAN (F.), note sous CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. contre world programming Ltd, RTD Com.*, 2012, p. 536.

133 En ce sens : CA Paris 23 janv. 1995 ; CJUE 2 mai 2012, *WPL c/ SAS Institute* ; pour une doctrine contraire : DERCLAYE (E.) et STROWEL (A.), « Droit d'auteur et numérique : logiciels, bases de données, multimédia », *loc. cit.*

134 Directive 91/250/CEE, *op. cit.*, art. 9.1 : « Les dispositions de la présente directive n'affectent pas les autres dispositions légales ».

135 Directive 2001/29/CE, 22 mai 2001, sur l'harmonisation de certains aspects du droit d'auteur et des droits voisins dans la société de l'information.

de ces titres, il serait protégé comme le programme lui-même¹³⁶.

48. – Pour une partie de la doctrine les langages de programmation ne sont qu'une théorie¹³⁷, un manuel non formalisé contenant les règles pour une programmation correcte¹³⁸, une sorte de manuel d'utilisation du compilateur lequel sera chargé de traiter ce code source. Considérer qu'un langage de programmation n'est, en tant que tel, qu'un « exercice intellectuel »¹³⁹ revient à proclamer son inexistence pure et simple. En effet, ce qui fait alors défaut, au-delà de toute appréciation de l'originalité, c'est la présence d'une forme. Or « le constat s'impose qu'il y a bien une écriture – une forme donc »¹⁴⁰, cette forme est fixée dans un écrit scientifique¹⁴¹. La formalisation se fait dans un document révélé à tous ceux qui souhaitent réaliser un logiciel de traduction¹⁴², ou bien gardé secret pour éviter la concurrence. Sur ce document sont décrits les équivalences et procédés de traductions du langage de programmation en code binaire.

L'existence formelle du langage s'appréciera dans les œuvres directement dérivées qui en découlent : le compilateur et l'interpréteur¹⁴³, l'éventuel environnement d'exécution, la bibliothèque contenant toutes les entrées possibles dans le langage considéré, le tutoriel ou manuel d'utilisation du langage. Pour monsieur l'avocat général Yves Bot, qui s'était prononcé à l'occasion de l'affaire *World Programming Ltd* contre *SAS Institute*¹⁴⁴ ; le langage est un mode d'expression et non l'expression elle-même. Il refuse donc à l'outil la protection du droit d'auteur. Cette position se défend, mais remet en cause la théorie de l'unité de l'art et s'oppose à la logique de la directive logiciel¹⁴⁵ qui protège les programmes d'ordinateur, eux-mêmes des œuvres-outils. Il faut faire abstraction des préjugés entourant l'objet qu'est le langage et appliquer un jugement neutre d'abord sur l'existence d'une forme, ensuite sur l'originalité¹⁴⁶.

Patrick Roussel admet que le langage de programmation soit une construction théorique, mais il le

136 Conformément aux articles 1^{er} et 2^e de la directive 91/250/CEE, *op. cit.*, et 1^{er} §2 de la directive 2009/24/CE, *op. cit.*

137 En particulier la syntaxe pour CHOMSKY (N.), *Le langage et la pensée*, Syntactic Structures, Paris, 1957.

138 Un mode d'emploi pour GRELON (B.), *Les entreprises de services*, *loc. cit.* ; un ensemble de méthodes pour ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *op. cit.*, n° 4, §5.

139 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *op. cit.*, n° 1, §4.

140 GAUDRAT (P.), *Le droit de l'informatique*, *op. cit.*, §3.

141 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *op. cit.*, n° 4, *in fine*.

142 On parlera de librairie, ou de « table de correspondance » selon : CHENIQUE (F.), *Software . Langages et systèmes d'exploitation*, Dunod, Paris, 1971, p. 29.

143 Qui sont les « instrument[s] de programmation mettant en œuvre un langage de programmation » selon : ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *op. cit.*, n° 1, §3.

144 BOT (Y.), *Conclusions de l'Avocat général*, CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. contre world programming Ltd*, points 68 à 76 des conclusions.

145 Directive 91/250/CEE, 14 mai 1991, concernant la protection juridique des programmes d'ordinateur.

146 MACREZ (F.), « Le droit d'auteur, le programme d'ordinateur et la cour de justice », *RIDA* 22 octobre 2012, titre I.B.2 « La confusion entre l'objet et le critère », §1. MACREZ (F.), « Le logiciel chez les Bretons (le retour) », *RLDI*, 1^{er} avril 2013.

classe parmi les œuvres scientifiques¹⁴⁷, donc potentiellement protégées par droit d'auteur. Il retient simplement qu'il s'agisse d'« une œuvre de peu de fantaisie », Christian Le Stanc le rejoint sur ce point, l'originalité sera complexe à caractériser¹⁴⁸, ce sera un travail au cas par cas¹⁴⁹.

49. – **Une jurisprudence subtile.** Rapprocher langage humain et langage de programmation est un exercice intéressant, ainsi pourrait-on se demander si un dictionnaire néologique et des phrases écrites dans un langage créé de toutes pièces sont ou non protégeables. On pense notamment à ceux créés par J.R.R. Tolkien dans son œuvre monumentale, ou encore l'espéranto. S'appuyer sur la jurisprudence *Griole*¹⁵⁰ qui a refusé la protection du vocabulaire utilisé dans un dictionnaire français-cajun ne paraît pas pertinent¹⁵¹. Aucune des deux langues n'a été créée par l'auteur qui s'est contenté de définir les termes et de rapprocher les équivalences linguistiques¹⁵². En revanche, la Cour d'appel de Paris admettrait qu'un langage de programmation puisse être une œuvre originale¹⁵³, « la reprise non contestée d'un langage mathématique existant n'est pas en soi suffisante pour conclure au défaut d'originalité »¹⁵⁴. Ils bénéficient de la protection du droit d'auteur à travers la reconnaissance de la manifestation de « l'originalité d'un langage théorique en langage permettant une application industrielle »¹⁵⁵. Elle raisonne sur un langage et sur le droit spécial applicable au logiciel. Les solutions étrangères adoptent une analyse inverse, la Cour britannique semble hostile à la protection des langages de programmation, pour s'économiser l'analyse de l'originalité elle conclut au défaut de forme¹⁵⁶.

50. – L'arrêt SAS de la Cour de justice de l'union européenne¹⁵⁷, est l'apport juridique majeur en matière de langage de programmation. Les juridictions britanniques étaient confrontées à un ensemble de logiciels développés par la société SAS et dont l'une des fonctionnalités était de permettre à l'utilisateur de créer de petits logiciels – que l'on appellera *scripts* – qui fonctionnaient

147 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *op. cit.*, n° 3, dernier §.

148 LE STANC (C.), « Droit de l'informatique », *La semaine juridique entreprise et affaires* n° 21, 23 mai 1996, 559, n° 6.

149 SIRINELLI (P.), *RIDA*, juil. 2012, p. 219.

150 CA Paris, 14 janv. 1992, *Griole c/ Vautrin*.

151 Mettant en perspective cette jurisprudence avec la question des langages de programmation sans en tirer de véritables conséquences et en admettant que « la question mérite d'être approfondie » : GAUTIER (P.-Y.), *Propriété littéraire et artistique*, Puf, 8^e éd., 2012, n° 130.

152 STÉRIN (A.-L.), *Guide pratique du droit d'auteur*, Maxima . Concrètement que dois-je faire, 2^e éd., 2011, n° 80.

153 Selon l'interprétation de : LE STANC (C.), « Droit de l'informatique », *loc. cit.* ;

154 CA Paris, 4^e ch, 10 nov 1994, PIBD 1995, 582, III, p. 27.

155 LAMZOUHRI (A.), *L'incidence de la protection du droit d'auteur sur le logiciel*, mémoire, 21 mai 2012, Paris-Ouest, accessible sur : <http://www.e-juristes.org/wp-content/uploads/2012/05/Lincidence-de-la-protection-du-droit-dauteur-sur-le-logiciel.pdf>.

156 Pour une analyse approfondie de la décision High court of Justice, Chancery Division, 25 janv. 2013, *SAS Institute Inc. vs. World Programming Limited*, cf. : MACREZ (F.), « Le logiciel chez les Bretons (le retour) », *RLDI*, 1^{er} avril 2013.

157 CJUE, 2 mai 2012, n° C-406/10, *SAS Institute Inc. c/ world programming Ltd*.

à travers ceux créés par la société SAS – que l’on appellera logiciel d’accueil. La création des *scripts* supposaient donc la reproduction du langage de programmation SAS, d’abord dans le logiciel d’accueil qui permet d’écrire les *scripts*, ensuite dans les *scripts* qui sont des œuvres dérivées de ce langage¹⁵⁸. Au point 39 de sa décision, la Cour de justice précise que « ni la fonctionnalité d’un programme d’ordinateur ni le langage de programmation et le format de fichiers de données utilisés dans le cadre d’un programme d’ordinateur pour exploiter certaines de ses fonctions ne constituent une forme d’expression de ce programme »¹⁵⁹. « Ce programme » désigne le logiciel d’accueil SAS dans le cadre duquel sont créés et exploités les *scripts*¹⁶⁰, comme la Cour le précise en son point 42¹⁶¹. La solution est logiquement¹⁶² à rapprocher de celle rendue concernant les interfaces graphiques¹⁶³. Imaginons un logiciel de dessin utilisant un système de tampons, chaque tampon est une image possiblement originale, et l’utilisateur, par l’association qu’il fera des différents tampons, pourra ainsi réaliser un dessin lui-même original. Les tampons, en tant qu’images, ne sont pas une forme d’expression du programme qui les met en oeuvre, il en est de même pour le langage SAS. Cela n’exclut pas qu’ils puissent être protégés par droit d’auteur de manière totalement indépendante, comme l’admet expressément la Cour de justice¹⁶⁴, ni même qu’un langage de programmation ne puisse être considéré comme un programme d’ordinateur¹⁶⁵.

La doctrine a été très troublée par cet arrêt, elle y a d’abord vu un simple rejet des langages de la protection par le droit d’auteur¹⁶⁶, puis a relevé l’ambiguïté¹⁶⁷, pour ensuite l’analyser comme une continuité de la jurisprudence sur les interfaces graphiques : certes la Cour écarte l’application du

158 Cf. *infra*, n° 95-100.

159 Souligné par nous.

160 En ce sens : MACREZ (F.), « Le droit d’auteur, le programme d’ordinateur et la cour de justice », *RIDA*, 22 octobre 2012, titre I.A.2 « La qualification distributive », §2.

161 CJUE, 2 mai 2012, *op. cit.*, point 42 : « Quant au langage de programmation et au format de fichiers de données utilisés dans le cadre d’un programme d’ordinateur pour interpréter et exécuter des programmes d’application écrits par les utilisateurs ainsi que pour lire et écrire des données dans un format de fichiers de données spécifique, il s’agit des éléments de ce programme au moyen desquels les utilisateurs exploitent certaines fonctions dudit programme. » (souligné par nous).

162 Procèdent au rapprochement : CJUE, 2 mai 2012, *op. cit.*, point 35, 37, 38 et 45 ; BERNAULT (C.), « Étendue de la protection des logiciels par le droit d’auteur », *L’essentiel. Droit de la propriété intellectuelle*, 15 juil. 2012, n°7, p. 2, §3. ; SIRINELLI (P.), *RIDA*, juil. 2012, p. 215, *in fine* ; MACREZ (F.), « Le droit d’auteur, le programme d’ordinateur et la cour de justice », *op. cit.*, « Introduction » §2, *in fine*.

163 CJUE, 22 déc. 2010, C-393/09, *Bezpenostní softwarová asociace* ; *Propriétés Intellectuelles*, 2011, 39, p. 205, note BENABOU (V.-L.) ; CCE, 2011, 5, comm. 42, CARON (C.).

164 En ce sens : CJUE, 2 mai 2012, *op. cit.*, point 45 : « *Le constat fait au point 39 du présent arrêt ne saurait affecter la possibilité pour le langage SAS et le format de fichiers de données de SAS Institute, de bénéficier, en tant qu’œuvres, de la protection par le droit d’auteur, en vertu de la directive 2001/29, s’ils sont une création intellectuelle propre à leur auteur.* ».

165 Cf. *infra*, n° 52-56.

166 En ce sens la majorité des brèves en ligne ainsi que : BERNAULT (C.), « Étendue de la protection des logiciels par le droit d’auteur », *loc. cit.* ; VIVANT (M.), « 160 - À propos des langages de programmation », *op. cit.*, partie mise à jour en mai 2012, §7.

167 MARINO (L.), « Chronique de jurisprudence de Propriété littéraire et artistique », *Gazette du Palais*, 2 août 2012, n° 215, p. 11, §2 : La cour « donne d’une main ce qu’elle vient de retirer de l’autre » en refusant la protection des langages par la directive de 1991 pour les protéger en vertu de la directive de 2001.

droit spéciale, mais admet une possible protection par le droit commun¹⁶⁸. Seuls quelques auteurs ont poussé plus loin l'analyse pour concéder que les langages puissent « constituer par eux-mêmes des œuvres logicielles originales protégeables »¹⁶⁹.

B) La véritable nature du langage : une programmation fondamentale à l'usage réglementé

51. – Un langage de programmation correspond, en réalité, à un programme à l'état d'œuf, une somme de briques et un ensemble cohérent qui constituent déjà des programmes (1). Les diverses possibilités d'assemblages en des séries d'instructions n'ont été rendues possibles que par l'écriture d'une véritable syntaxe dirigeant la combinaison des instructions en séries, la règle d'utilisation est un ensemble complexe protégeable (2).

1. Le langage : un programme préparatoire dans la conception de logiciels

52. – Pour Pierre-Yves Gautier un langage de programmation pourrait être un « type de programme »¹⁷⁰, mais en vertu de la directive programmes d'ordinateurs de 1991, le langage ne pourra être considéré comme un programme d'ordinateur que dès lors qu'il sera démontré qu'il s'agisse d'une série d'instructions ou bien d'une étape préparatoire dans la conception d'un autre programme.

53. – **Le langage : un programme à part entière ?** Un langage peut être considéré comme une série d'instructions. En réalité, c'est d'abord une compilation de mots dont chacun renvoi à une série d'instructions écrite dans un langage de programmation intermédiaire ou directement en binaire¹⁷¹. D'après le professeur Caron « L'originalité du dictionnaire résidera souvent dans l'expression des définitions »¹⁷². Nous ne disons pas que la signification du mot soit protégeable, il s'agit d'une fonctionnalité aussi complexe soit-elle, mais que sa description l'est ; c'est-à-dire la série d'instructions qui est la traduction du terme d'un langage dans un langage moins évolué¹⁷³. Il s'agit de distinguer ce que l'on appelle une « fonction », la suite d'instructions liée à un mot du langage, de la fonctionnalité. On notera par ailleurs qu'il est courant, en matière de programmation, d'élaborer des sous-programmes, des modules, auxquels on fera référence dans le code principal par

168 En ce sens : SIRINELLI (P.), *RIDA*, juil. 2012, p. 215-221 ; MACREZ (F.), « Le droit d'auteur, le programme d'ordinateur et la cour de justice », *op. cit.*, titre I « La méthode de qualification ».

169 BOUCHE (N.), Commentaire de la décision CJUE 2 mai 2012, Propriété industrielle n° 7, juil. 2012, comm. 61, n° 1, *in fine* ; *cf. infra* n° 52-56.

170 GAUTIER (P.-Y.), *Propriété littéraire et artistique*, Puf, 8^e éd., 2012, n° 130.

171 *Cf. supra.*, n° 38-Erreur : source de la référence non trouvée.

172 CARON (C.), *Droit d'auteur et droits voisins*, Litec . LexisNexis, 2^e éd., 2009, n° 128.

173 *Cf. supra*, n° 14.

l'emploi du nom du fichier correspondant au sous-programme et de son adresse, on est ici très proche de notre sujet d'étude. D'ailleurs, monsieur Dréan utilise la notion de « macro-instruction » pour désigner les modules écrits lors des premières étapes d'écriture d'un programme¹⁷⁴, tandis que monsieur Chenique l'utilise pour désigner les termes d'un langage évolué¹⁷⁵.

Ensuite, cette compilation d'instructions peut elle-même constituer un programme. « *Un logiciel est une série d'instructions en un langage informatique (langage machine, par exemple) destinées à faire exécuter par un ordinateur (ou calculateur, ou microprocesseur) des opérations définies par un algorithme, représenté par un "ordinogramme", ou organigramme* »¹⁷⁶. La définition du logiciel distingue la *série d'instructions* du *suivi de la ligne directrice tracée dans l'organigramme*, le logiciel est la réunion de ces deux éléments. On sait que les langages de programmation suivent un algorithme, lequel est fondamental dans les langages les plus évolués comme le C++¹⁷⁷, et permettent parfois l'économie du calcul d'un algorithme lors de la conception d'un logiciel. Une série d'instructions est-elle forcément une suite logique et exécutable d'instructions, un logiciel, ou peut-on se contenter d'un ensemble cohérent d'instructions, la bibliothèque ou le dictionnaire d'un langage de programmation ? Il est vrai que l'article 1 de la directive de 1991 définit les programmes d'ordinateur comme étant des œuvres littéraires, or un recueil est une œuvre littéraire. Il nous semble pourtant que la notion même de programme d'ordinateur suppose une suite logique, linéairement ou alternativement. La cohérence du langage, le suivi d'un algorithme, pour proposer une programmation fondamentale pourrait ne pas suffire pour qualifier l'intégralité du langage de programme d'ordinateur. Pour autant, chaque élément lexical pourrait indépendamment être considéré comme une suite d'instructions, un mini-programme, dès lors que la série d'instructions en binaire, ou dans un langage intermédiaire, aura été suffisamment développée.

54. – Non seulement le terme d'un langage peut être individuellement considéré comme une suite d'instructions, mais le langage en son entier peut être considéré comme une étape préparatoire à la conception d'un logiciel. La notion de macro-instruction permet de justifier l'analyse à l'égard de chaque mot d'un langage, il convient de l'étendre au langage lui-même. La théorie de la double face du langage de programmation, développée par Philippe Gaudrat, conforte notre position. Pour cet auteur les langages sont conçus pour être tournés vers l'homme et tournés vers la machine¹⁷⁸. Il

174 DRÉAN (G.), *L'industrie informatique . Structure, économie, perspective*, Masson, 1996, p. 214.

175 CHENIQUE (F.), *Software . Langages et systèmes d'exploitation*, Dunod, Paris, 1971, p. 29-36.

176 MARTIN (J.-P.), « La protection des logiciels informatiques : droit d'auteur ou brevet d'invention ? », *La Semaine Juridique Entreprise et Affaires* n° 16, 25 avr. 1990, 15752, §9.

177 GRENIER (J.-G.), *Dictionnaire d'informatique et d'internet . Anglais-français*, La maison du dictionnaire, 1996, v° « *Programming language* », p. 521 : « *Algorithms are as important in C++ as they are in any other programming language* » traduction (par nous) : « Les algorithmes sont aussi importants en C++ que dans tout autre langage de programmation ».

178 GAUDRAT (P.), « L'invention informatique : un débat difficile et contourné », *RTD Com.* 2005, p. 512, n° 6-9.

s'agit en fait, selon nous, d'une programmation fondamentale de séries d'instructions qui pourront, au grès du choix du développeur, être assemblées en suites linéaires ou alternatives d'instructions. Les langages sont donc des étapes permettant à terme le développement de programmes, leur caractère indispensable¹⁷⁹ renforce cette idée d'étape préparatoire à la conception non pas d'un unique programme d'ordinateur, mais d'une pluralité.

55. – **Le langage : une étape préparatoire dans la réalisation d'un programme.** Pour la Cour de justice, « l'objet de la protection de la directive 91/250 englobe les formes d'expression d'un programme d'ordinateur ainsi que les travaux préparatoires de conception susceptibles d'aboutir, respectivement, à la reproduction ou à la réalisation ultérieure d'un tel programme »¹⁸⁰. Le langage permet bien d'aboutir à la « réalisation ultérieure d'un programme », sans quoi il ne serait pas tant indispensable à leur réalisation. Écartons d'emblée la confusion, le langage n'est pas le matériel de conception préparatoire des instruments de programmation¹⁸¹, compilateur et interpréteur qui le mettent en œuvre¹⁸², mais de conception préparatoire de tous les logiciels qu'il va permettre d'écrire. Cependant, peut-être exige-t-on des travaux de conception préparant spécialement la réalisation d'un programme d'ordinateur défini. On remarquera qu'à une époque les langages de programmation étaient créés pour l'élaboration d'un unique programme, la pratique à évoluer et les programmeurs se contentent d'élaborer des variantes ou des aménagements de langages déjà existants¹⁸³. Il est aussi courant d'écrire de nouvelles « fonctions » ou sous-programmes sous la forme de petites suites d'instructions, de les enregistrer dans un « fichier » à part et d'y renvoyer dans le « fichier » principal du programme.

La cour de justice, dans le point 39 de son arrêt du 2 mai 2012, se contente de remarquer que le langage SAS n'est pas un mode d'expression du logiciel SAS. Que cela puisse signifier que le langage SAS puisse être un mode d'expression d'un autre logiciel ne nous paraît pas pertinent. Au contraire le langage SAS pourrait être en lui-même un programme d'ordinateur en ce qu'il est une série d'instructions cohérente, mais aussi en ce qu'il est une étape préparatoire à la conception d'autres programmes. Nous l'avons vu, l'emploi du terme « ce » au lieu de la combinaison « d'un »

179 Cf. *supra*, n° 11-15 et 28-30.

180 CJUE, 2 mai 2012, n° C-406/10, *SAS Institute Inc. c/ world programming Ltd*, point 37.

181 Se servant de l'argument pour écarter la protection du langage en tant que matériel de conception préparatoire : ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *CCE*, n° 4, avril 2005, étude 15, n° 6, §4 et 5.

182 Cf. *infra*, n° 97.

183 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *op. cit.*, n° 1, §3 : « Alors que, pendant longtemps, l'usage du langage de programmation s'est principalement limité au cercle restreint des bureaux d'études et de développement de programmes d'ordinateur pour ne plus servir après leur mise au point sous forme de code objet, il n'en est plus ainsi aujourd'hui. »

n'est pas dénuée d'intérêt. Par ailleurs, le langage n'est pas un mode d'expression du logiciel-compilateur de SAS, ce sont plutôt les logiciels écrits en langage SAS qui sont des modes d'expression du langage dans lequel ils sont écrits. De plus, nous le verrons, les compilateurs et interpréteurs sont les modes d'expression les plus complets des langages de programmation¹⁸⁴.

56. – Il faut considérer que la Cour de justice ne rejette pas l'application du droit spécial applicable aux programmes au profit du droit commun du droit d'auteur¹⁸⁵. Pourtant, la Cour fait expressément référence à la directive de 2001 en admettant dans son point 42 que les remarques faites au point 39 ne font pas obstacle à la protection des langages de programmation par le droit d'auteur. Il pourrait s'agir d'un renvoi impératif au droit commun du droit d'auteur, ou simplement d'un renvoi à la directive la plus générale, une affirmation de la non-exclusion de principe des langages de programmation du droit d'auteur. La première hypothèse nous attire spontanément, mais la seconde a notre préférence, puisque les langages sont constitués d'éléments divers, dont certains ont des régimes spéciaux : la base de données et les suites d'instructions.

Une autre hypothèse est envisageable, la cour ne s'intéresserait pas aux éléments du langage dont la protection est acquise, telle que les séries d'instructions, les néologismes ou les bases de données, mais à l'élément central du langage de programmation : sa syntaxe.

2. *La syntaxe : une règle conditionnant l'utilisation*

57. – Ce qui marque le plus la spécificité d'un langage ce n'est pas le dictionnaire, éventuellement néologique, mais plutôt l'ensemble des règles qui le régissent. La réflexion ne se limite pas aux seuls langages de programmation, car avant d'être destiné à la programmation, il s'agit d'un langage. L'irrespect d'une règle d'un langage humain peut occasionner des difficultés de compréhension, pour notre sujet l'irrespect rend inexécutable le programme. Une telle règle d'utilisation doit-elle être protégée par le droit d'auteur ? Il est possible d'envisager, indépendamment, une protection de la syntaxe, soit en considérant qu'il s'agisse du cœur du langage de programmation, soit en considérant par analogie qu'il s'agisse d'une œuvre normative.

58. – **Protection de la syntaxe comme cœur du langage de programmation.** Il ne serait pas pertinent de considérer que les langages de programmation ne soient que des œuvres composites. La protection des éléments du langage précédemment étudiés est acquise depuis longtemps, qu'il s'agisse des séries d'instructions, des néologismes ou de la base de données. Même analyse

184 Puisqu'ils les reproduisent intégralement : *cf. infra*, n° 97.

185 SIRINELLI (P.), note sous Cour de Justice de l'Union Européenne (CJUE), grande Chambre, 2 mai 2012, affaire numéro C-406/10, SAS Institute Inc. contre World Programming Ltd, *RIDA*, juil. 2012, p. 213 ; MACREZ (F.), « Le droit d'auteur, le programme d'ordinateur et la cour de justice », *RIDA*, 22 octobre 2012, §7-8.

concernant l'œuvre composite que forme alors le langage. Dans le cas contraire les langages de programmation seraient, à l'instar des œuvres multimédias, de simples créations doctrinales correspondant à des œuvres composites sans aucune spécificité qui leur serait propre. Or les langages suivent chacun leur fil directeur, une syntaxe propre, pour constituer un mode de communication intelligible. Or, l'arrêt SAS du 2 mai 2012 prévoit la protection des langages de programmation dès lors qu'ils sont originaux et ce par application de la directive de 2001¹⁸⁶, soit le droit commun du droit d'auteur. Il est donc fait abstraction d'une partie substantielle de l'œuvre composite : ses séries d'instructions. Il est probable que la Cour ne vise pas les éléments accessoires du langage mais la partie qui en constitue le cœur, la syntaxe. Les juges de Luxembourg offrent alors une lumière intéressante sur le sujet. L'argument sera évoqué et développé¹⁸⁷ pour contester le régime distributif issu de la jurisprudence *Cryo*¹⁸⁸.

Il faut cependant nuancer le propos, d'une part, car dans beaucoup de langages la syntaxe aura une importance assez faible par rapport aux traductions du lexique, les séries d'instructions, d'autres part, car l'objectif utilitaire conditionne, ici plus qu'ailleurs, le choix du concepteur. C'est donc l'originalité qui risque de faire défaut. On peut, toutefois, étudier le degré d'exigence ainsi que le siège de l'originalité des textes normatifs pour procéder par analogie.

59. – **Protection d'une syntaxe par analogie avec les textes normatifs.** De nombreuses décisions ont été rendues, à propos de clauses contractuelles, de règles du jeu, de normes de certification et de textes officiels. Une syntaxe semble se rapprocher d'une part des normes de certification du fait de son autonomie, d'autre part des règles d'un jeu du fait de sa dépendance du reste du langage de programmation.

60. – André Bertrand relève qu'il existe des normes AFNOR obligatoires et d'autres facultatives¹⁸⁹. Cependant, il nous semble que seules les premières ne bénéficieront pas du monopole du droit d'auteur¹⁹⁰. L'AFNOR elle-même se réserve ses droits d'auteur sur ses normes. Mais que réserve-t-elle en réalité : chaque norme, l'ensemble de normes ou la formalisation de la norme ?¹⁹¹ Protéger la norme conduirait à une réservation de l'idée et doit être strictement rejetée,

186 Directive 2001/29/CE, 22 mai 2001, sur l'harmonisation de certains aspects du droit d'auteur et des droits voisins dans la société de l'information.

187 Cf. *infra*, n° 75.

188 Cass. Civ. 1^{re}, 25 juin 2009, n° 07-20.387, *Cryo*.

189 BERTRAND (A.R.), *Chapitre 202 logiciels in Droit d'auteur*, éd. Dalloz, 2013, n° 202.26.

190 Cf. concernant la disparition du monopole du droit d'auteur sur des normes AFNOR devenues obligatoires : FROCHOT (D.), « Du nouveau sur les normes et sur l'AFNOR », Les infostrateges [en ligne], 26 juin 2009, <http://www.les-infostrateges.com/actu/0906742/du-nouveau-sur-les-normes-et-sur-l-afnor>, consulté le 5 mai 2013.

191 Pour une étude sur la question cf. CHAMPIGNEULLE-MIHAILOV (J.), « Les conflits relatifs à l'utilisation des normes par les professionnels », Petites affiches, 11 février 1998, n° 18, p. 21.

peu importe le degré d'élaboration de celle-ci. En revanche, l'ensemble des normes constitue une base de données, ou un recueil de normes. Est alors protégé « le choix ou la disposition des matières » de cette base de données en vertu de l'article L. 122-3 du Code de la propriété intellectuelle, nous l'avons vu¹⁹². Enfin, la formalisation de chacune des normes doit être protégée. Les juridictions ont notamment admis la réservation de conditions générales de vente sur ces fondements¹⁹³. Ainsi le fond est libre et la forme est protégée. Peu importe que l'idée ait un objectif fonctionnel ou un objectif ludique, l'idée est de libre parcours comme l'a rappelé la Cour d'appel de Paris dans l'affaire *Karbone 14*¹⁹⁴. C'est le défaut de forme qui est reproché à la règle du jeu si l'on en croit Jean-Michel Brugière¹⁹⁵.

61. – Pour autant, le choix de la règle associée au reste du langage de programmation afin d'en gérer le fonctionnement ne pourrait-il constituer une forme originale ? C'est alors la symbiose entre le fond de la règle et l'élément dont elle gère l'utilisation qui serait protégée. On se rapprocherait ainsi de l'affaire *Monopoly*¹⁹⁶ : ont été protégés comme marque figurative la règle d'un jeu associée au plateau de jeu, comme étant les éléments les plus significatifs d'un jeu. En réalité, ce n'est pas le *design* du plateau, ou la forme donnée au texte matérialisant la règle en tant qu'œuvre littéraire qui doit être protégé, mais bien la règle en tant qu'elle se formalise directement dans le jeu lui-même et qu'elle en conditionne l'utilisation. On peut même se demander si le plateau de jeu n'est pas une autre forme de règle du jeu en ce qu'il en définit les limites matérielles, ainsi que la diversité des coups possibles. Il n'est pas possible de déplacer son pion en dehors de la surface. De la même façon, il n'est pas possible dans un jeu-vidéo de faire exécuter des mouvements qui n'auraient pas été définis dans le *game play* et formalisés dans le code. En matière de langages de programmation ce serait donc la syntaxe (la grammaire) en ce qu'elle se formalise au sein de celui-ci et en ce qu'elle forme un tout avec les limites matérielles du langage – c'est-à-dire le lexique (le dictionnaire) – qui devrait faire l'objet de la protection.

Développons l'idée sous l'angle des jeux-vidéos. Depuis un arrêt *Cryo*¹⁹⁷, un jeu-vidéo doit être considéré comme étant une œuvre composite pour laquelle il faut appliquer le droit d'auteur correspondant à chacun de ses éléments¹⁹⁸. Pourtant, il nous semble bien qu'avant d'être vidéo le jeu électronique est un jeu dont le cœur est constitué de la règle et du *game play*. Les procédés

192 Cf. *supra* n° 42.

193 CA Paris, 4e ch., 24 sept. 2008.

194 CA Paris P. 6, ch. 1, 17 oct. 2012 ; com. par : BRUGIÈRE (J.-M.), in « Droit d'auteur et droits voisins » (chronique), Propriétés intellectuelles n° 46, janvier 2013, p. 43-44.

195 BRUGIÈRE (J.-M.) com. sur CA Paris P. 5, ch. 1, 17 oct. 2012 in « Droit d'auteur et droits voisins » (chronique), Propriétés intellectuelles n° 46, janvier 2013, p. 43-44.

196 CA Paris, 4e ch., 7 mars 2007, n° 06/02813, *Monopoly*.

197 Cass. civ. 1^{re}, 25 juin 2009, n° 07-20.387, *Cryo*.

198 Cf. *infra*, n° 66.

logiciels ne servent qu'à sa mise en œuvre. La virtualisation peut aisément se passer de programmes informatiques, prenons l'exemple du *babyfoot* ou du plateau de jeu du *monopoly*. Excepté la partie logicielle il y a bien là tous les ingrédients des jeux-vidéos : des graphismes, un scénario, une interactivité directement avec le jeu et indirectement avec le ou les adversaires. La spécificité d'un jeu ne réside pas non plus dans ses graphismes ou éléments audiovisuels, il y a des jeux-vidéos dont l'interface est composée d'une succession de pages internet¹⁹⁹. En matière de jeux classiques l'exemple phare est celui des jeux de rôle²⁰⁰. Comme l'a admis le Conseil supérieur de la propriété littéraire et artistique, la spécificité d'un jeu réside en réalité dans ses règles et son *game play*²⁰¹, il s'agit des règles qui conditionnent le comportement du joueur et les modalités de l'interactivité.

Les jeux-vidéos et les langages de programmation auraient pour cœur la règle associée à une sorte de *game play*, c'est-à-dire un ensemble de limites matérielles qui conditionnent impérativement le joueur et le programmeur. L'un et l'autre peuvent prendre le risque d'enfreindre la règle en espérant que cela passe inaperçu, que le programme fonctionne ; cependant, la contrainte matérielle, elle, est infranchissable. En réalité ces deux contraintes ont une relation fusionnelle, en fonction du degré d'élaboration de la syntaxe, celle-ci définirait même des règles spéciales de traduction selon la combinaison des termes, contrainte matérielle, et pourrait elle-même être considérée comme une partie de la série d'instructions²⁰².

62. – Ainsi, au-delà des apparences, un langage de programmation est éligible à la protection par le droit d'auteur, et ce pour plusieurs de ses éléments : les éléments et l'ensemble que constitue le dictionnaire éventuellement néologique ; la fiche descriptive du langage ; la syntaxe en tant que règle ou en tant que cœur du langage de programmation ; les séries d'instructions qui sont les traductions du langage établies par son concepteur ; le langage lui-même en tant que programme ou en tant qu'étape dans la conception d'un logiciel. La solution est bienheureuse au regard des efforts considérables que leur conception nécessite, d'autant plus que les arguments opposés à leur protection pourraient être étendus à l'ensemble des créations utilitaires. Il convient maintenant de tirer les conséquences d'une telle protection par le droit d'auteur pour envisager le régime applicable aux langages de programmation et l'influence de ce régime sur les œuvres qui utilisent les langages de programmation.

199 Par exemple le jeu *O-Game* dans lequel les seuls graphismes sont de simples illustrations dont l'importance est accessoire, www.ogame.fr, consulté le 28 mai 2013.

200 Dans le jeu emblématique *Donjon et Dragons*, chaque joueur crée son propre personnage et seule importe la règle du jeu et l'imagination des joueurs et du maître de jeu.

201 CSPLA avis n° 2005-1, 7 déc. 2005.

202 Cf. *infra* n° 53.

Chapitre 2 : Le régime juridique complexe des langages de programmation

63. – La protection des langages de programmation au titre du droit commun et du droit spécial du droit d’auteur influence directement le régime applicable qui sera distributif, alors qu’un régime unitaire préférable doit être recherché (I), ce qui ne simplifiera pas l’exploitation multiforme des langages de programmation qui devra être raisonnée (II).

I) D’un régime distributif vers un régime unitaire préférable

64. – Si la jurisprudence impose une qualification éclatée contestable des éléments d’un langage de programmation (A), des mécanismes juridiques permettent d’atteindre un régime unitaire préférable (B).

A) Une qualification éclatée contestable

65. – Par parallélisme avec les œuvres multimédias, les langages de programmation bénéficient d’une qualification éclatée, comprenant à la fois des éléments logiciels et des éléments soumis au droit commun du droit d’auteur (1), mais tout comme pour ces premières, cette qualification est juridiquement contestable ou, du moins, à compléter (2).

1. Principe de l’éclatement et distribution mal aisée

66. – **Un régime distributif de principe.** Le parallèle avec l’œuvre multimédia est pertinent puisque les deux types d’œuvres correspondent à des compositions d’une multitude d’éléments juridiquement distinguables, mais assemblés de manière cohérente. La doctrine fut longtemps partagée entre le courant unitaire et le courant distributif, en particulier concernant les jeux-vidéos. Certains y voyaient une œuvre purement logicielle²⁰³ et d’autres une œuvre audiovisuelle²⁰⁴, d’autres encore une base de données en ce que divers éléments de nature différente étaient assemblés dans une même œuvre. Cette dernière hypothèse doit être strictement rejetée, la réflexion fait abstraction de la cohérence avec laquelle est composée l’œuvre multimédia, le jeu ou le langage de programmation ; elle pourrait même être transposée à toute œuvre composite voire à toute œuvre, un film serait une base de données d’images et de sons, un livre serait une base de données de

203 TREPPOZ (É.), « La qualification logicielle d’un jeu-vidéo : un modèle pour les œuvres multimédia », *Les petites Affiches*, 230, 18 nov. 1999, p. 10.

204 LATREILLE (A.), *Les mécanismes de réservation et les créations multimédias*, thèse soutenue le 20 déc. 1995, Paris-Sud, p. 592 et s. ; analyse rejetée par la Cass. civ 1^{re}, 28 janv. 2003, *Casariil c/ Société Havas interactive*.

paragraphes. S'inspirer du régime de l'œuvre audiovisuelle est intéressant pour les jeux-vidéos, mais certains jeux disposent de graphismes particulièrement rudimentaires, certains ne disposent pas du moindre élément audiovisuel, ou bien ceux-ci n'ont qu'une place accessoire dans le jeu²⁰⁵. Les premiers jeux-vidéos, plus exactement jeux électroniques, étaient fournies non sur support magnétique, mais sur papier. Il fallait ainsi entrer une à une les lignes de code source avant de pouvoir utiliser l'œuvre « multimédia », le jeu était alors pratiquement du « tout logiciel », exactement l'opposé de la thèse du professeur Latreille qui prédisait la disparition des logiciels dans les jeux-vidéos au profit des seuls graphismes et élément audiovisuels²⁰⁶. On pourrait voir la spécificité de l'œuvre multimédia dans son absence de linéarité et dans son interactivité, ainsi que dans l'objectif de distraction. Il faudrait donc dégager un régime unitaire nouveau. Néanmoins, l'interactivité se retrouve dans tout logiciel destiné à l'utilisateur. Si tant est que l'on puisse qualifier un jeu classique d'œuvre de l'esprit²⁰⁷, on ne peut non plus ignorer l'interactivité et l'objectif de distraction qui existe dans un jeu de carte, un sport collectif ou un jeu de plateau.

67. – En 2009 la première chambre civile de la Cour de cassation rendait un arrêt *Cryo* qui offre une solution à la portée extrêmement large. Selon celui-ci, il faut appliquer à chaque élément d'une œuvre multimédia le droit d'auteur qui lui correspond. Cette jurisprudence s'ancre en totale opposition avec la doctrine unitaire, mais ne nous semble répondre qu'à une partie de la problématique. Il était alors question d'un jeu-vidéo et du régime de rattachement de celui-ci, il ne doit pas être considéré comme principalement logiciel et accessoirement agrémenté de graphismes, de musiques, de texte et d'une base de données.

68. – **La distribution des régimes.** Comme le proposent messieurs Macrez²⁰⁸ et Sirinelli²⁰⁹ en admettant le double régime droit spécial-droit commun, il faut procéder à une extension de la jurisprudence *Cryo*²¹⁰ à notre sujet d'étude qui est composé d'une multitude d'éléments, notamment des formes programmées. La Cour de Justice invitait à le faire en matière d'interface graphique dans son arrêt de décembre 2010²¹¹ et systématise désormais le principe d'éclatement dans l'arrêt

205 À titre d'exemple le jeu en ligne *O-Game* qui ne requiert que l'usage du navigateur sans même utiliser de *JAVA*, à une époque ; ou des jeux dans lesquels le graphisme n'a qu'une importance marginale comme le sudoku.

206 LATREILLE (A.), « La création multimédia comme œuvre audiovisuelle », *La Semaine Juridique*, éd. Générale, n° 31, 29 juil. 1998, I 156.

207 *Cf. supra* n° 61.

208 MACREZ (F.), « Le droit d'auteur, le programme d'ordinateur et la cour de justice », *RIDA*, 22 octobre 2012.

209 SIRINELLI (P.), « Propriété littéraire et artistique », *Recueil Dalloz* n° 42, 2012, p. 2836 ; SIRINELLI (P.), note sous Cour de Justice de l'Union Européenne (CJUE), grande Chambre, 2 mai 2012, affaire numéro C-406/10, *Institute Inc. contre World Programming Ltd*, *RIDA*, juil. 2012, p. 215.

210 Ayant tranché la qualification des œuvres multimédia, y compris des jeux-vidéos, en admettant un régime distributif : Cass. civ. 1^{re}, 25 juin 2009, n° 07-20.387, *Cryo*.

211 CJUE, 22 déc. 2010, C-393/09, *Bezpečnostní softwarová asociace*.

SAS de juin 2012²¹².

69. – Sans difficulté, c'est le droit spécial de la directive de 1991 et de 2009 qui s'appliquent aux séries d'instructions auxquelles renvoi le langage de programmation²¹³. Il en est de même si l'ensemble du langage devait être considéré comme une étape préparatoire dans la conception d'un programme d'ordinateur²¹⁴ ou comme un programme en lui-même²¹⁵.

70. – C'est le droit commun du droit d'auteur qui s'applique aux néologismes, au lexique. Le droit commun devrait aussi s'appliquer à la syntaxe dans sa résonance avec le lexique. Il faudrait pour cela considérer que lorsque la Cour de justice énonçait que les langages de programmation pouvaient faire l'objet d'une protection par le droit d'auteur en vertu de la directive 2001/29²¹⁶, en fait elle visait le cœur des langages de programmation, la syntaxe²¹⁷. C'est encore le droit commun du droit d'auteur qui s'applique au manuel d'utilisation du programme, précise la Cour de justice dans l'arrêt SAS. Il doit en être de même au tutoriel expliquant, décrivant le fonctionnement du langage ou la façon de l'utiliser.

71. – Le régime applicable à la bibliothèque du langage de programmation est plus complexe. Il s'agit a priori d'une base de données, ou bien d'un recueil de mots – néologiques ou non – dont l'originalité résidera davantage dans le choix des matières que dans leur disposition. Contrairement à un dictionnaire normal, on ne pourra reprocher le choix de l'exhaustivité, puisque les mots existants dans le lexique d'un langage de programmation procèdent, quant à eux, du choix du concepteur. La difficulté étant de savoir si le choix des matières est soumis au droit commun du droit d'auteur ou au droit spécial des bases de données. L'arrêt *Football dataco* de la Cour de justice²¹⁸ n'aidera pas à la compréhension, puisqu'il étend le droit d'auteur spécial des bases de données au fond de recherche, soit le choix des matières, alors qu'il devrait être réservé à la seule structure, la disposition des matières. Or, seul le choix des matières présente un intérêt en ce qui concerne directement les langages de programmation. Il subsiste donc une incertitude sur ce point, incertitude qu'il faut tempérer, car le droit spécial des bases de données diffère bien peu du droit commun.

72. – S'il faut étendre le mécanisme dégagé dans l'arrêt *Cryo*, il faut, par la même, appliquer une réserve vis-à-vis de cette vision distributive.

212 CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. c/ world programming Ltd.*

213 Cf. *supra* n° 52-53.

214 Cf. *supra* n° 55-56.

215 Cf. *supra* n° 54.

216 Directive 2001/29/CE, *Loc. cit.*

217 Cf. *supra* n° 58.

218 CJUE, C-604/10, 1^{er} mars 2012, *Football dataco*.

2. Qualification recentrée sur l'œuvre composite ou le cœur de l'œuvre

73. – Bien que nous saluions la solution de l'arrêt *Cryo*²¹⁹, plus juste et plus proche des réalités que les propositions de régimes unitaires, il nous faut soulever des incohérences qui prêchent pour l'unité. Si l'on suit l'arrêt *Cryo*, une œuvre multimédia est en fait une œuvre composite qui doit être décomposée pour qu'à chacun de ses éléments soit appliqué un régime propre, c'est donc faire abstraction de la qualité d'œuvre composite ainsi que des spécificités de certaines œuvres composites.

74. – **Y a-t-il abstraction de la qualité d'œuvre composite ?** L'article L. 113-3 du Code de la propriété intellectuelle²²⁰ semble bien renvoyer au droit commun du droit d'auteur, certes « sous réserve des droits de l'auteur de l'œuvre préexistante » ce qui pourrait ne pas signifier qu'au sein de l'œuvre global chaque élément vive sa vie indépendamment. Ainsi en tant qu'il est incorporé dans l'œuvre composite, la forme programmée devrait se soumettre au droit commun du droit d'auteur. Seulement l'incorporation elle-même supposerait le respect des prescriptions du droit spécial. Le point fort de l'analyse est qu'elle ne contredit pas la solution de l'arrêt *Cryo* dans lequel il s'était s'agit de faire respecter les droits de la SACEM sur des musiques diffusées au sein d'un jeu-vidéo, mais son point faible réside dans son application quasiment systématique aux logiciels composés de graphismes, soit la majorité des programmes informatiques. Cela ne laisserait qu'une place marginale au droit spécial, par ailleurs chaque élément du tout est, en réalité, créé dans la finalité de constituer un ensemble ; la finalisation d'un élément n'est qu'une étape de création plutôt que l'apparition d'une œuvre préexistante à l'œuvre finale. Il faudra davantage réfléchir en termes d'œuvre collective ou de collaboration²²¹.

75. – **Y a-t-il abstraction de la spécificité des jeux-vidéos et des langages de programmation ?** C'est la même spécificité qui prime au sein des langages de programmation et des jeux-vidéos, à savoir les règles d'utilisation²²², tous les autres éléments n'étant qu'accessoires ou bien leur défaut d'originalité est indifférent à la qualification²²³. L'arrêt *Cryo* pose en principe la qualification éclatée sans qualifier ce qui nous semble constituer le cœur même du jeu-vidéo, et du langage de programmation, la règle et le *game play*. Une étude approfondie de l'arrêt *Bezpečnostní softwarová asociace* de décembre 2010 permet d'aller vers une analyse inverse. Comme le relève

219 Cass. civ. 1^{re}, 25 juin 2009, n° 07-20.387, *Cryo*.

220 « L'œuvre composite est la propriété de l'auteur qui l'a réalisée, sous réserve des droits de l'auteur de l'œuvre préexistante. »

221 Cf. *infra*, n° 90-91.

222 Ce qui justifierait sa protection par le droit d'auteur, cf. *supra*, n° 58 et 61.

223 Cf. *supra*, n° 45-46.

monsieur Macrez²²⁴, la Cour de justice n'est pas seulement incorrecte dans sa qualification, mais va jusqu'à se contredire dans son arrêt de 2010. Elle est incorrecte puisque, contrairement à ce qu'elle semble avancer, on ne saurait soutenir qu'une interface graphique n'est que pur graphismes, soumis au droit commun du droit d'auteur. Il y a forcément une partie logicielle qui permet la mise en œuvre de l'interface, des suites d'instructions pour positionner les éléments sur l'écran, pour rendre efficace le clic de l'utilisateur, etc. En général, tout un pan du programme est exclusivement dédié à cette interface. Elle se contredit puisqu'elle relève dans un premier temps que l'interface graphique n'est pas un mode d'expression du programme²²⁵, qu'il faut donc lui appliquer le droit commun du droit d'auteur²²⁶. Elle affirme dans un deuxième temps que la télédiffusion ne met pas en œuvre le droit d'auteur sur l'interface graphique, puisque les utilisateurs ne peuvent pas utiliser l'interface mais simplement la voir²²⁷. Or, c'est le programme que l'on utilise et les graphismes que l'on voit.

En d'autres termes, l'interface graphique est soumise au droit commun du droit d'auteur, mais il n'y a représentation de l'interface que si l'utilisateur peut lui-même utiliser le logiciel. Trois solutions : soit il faut considérer que les juges de la Cour de justice n'ont pas correctement analysé à la fois le fonctionnement d'une interface graphique et le droit d'auteur mis en œuvre dans cette espèce ; soit les juges de Luxembourg rejettent la solution distributive française en appliquant à l'interface dans son entier – forme graphique et forme programmée – le droit commun du droit d'auteur ; soit la Cour de justice a fait création prétorienne d'un nouvel objet au sein de l'interface graphique. Il faut rejeter le premier raisonnement, opéré par l'absurde, pour conserver le second et en particulier le dernier qui est compatible avec la jurisprudence *Cryo*. Selon nous, cet arrêt dégagerait un troisième objet de droit d'auteur au sein de l'interface graphique : il y avait les graphismes, il y avait le programme, il y a désormais les modalités d'interactivité telles qu'elles sont ressenties lors de l'utilisation et qui sont soumises au droit commun du droit d'auteur²²⁸. Certains objecteront qu'un tel élément relève du domaine des idées. Ne s'agit-il pas de fonctionnalités ? Si en tant que telles les fonctionnalités ne peuvent être protégées²²⁹, l'ensemble constitué de fonctionnalités disposées d'une manière ou d'une autre – un menu ou l'ordre des icônes par exemple – peut certainement être protégé par droit d'auteur dès lors qu'il y a eu formalisation originale au sein d'un logiciel.

224 MACREZ (F.), « Le droit d'auteur, le programme d'ordinateur et la cour de justice », *RIDA*, 22 octobre 2012, titre I.B.1 : « la confusion est telle qu'elle confine à l'absurde ».

225 CJUE, 22 déc. 2010, C-393/09, *Bezpečnostní softwarová asociace*, points 34, 37 et 41.

226 *Idem*, points 44-46.

227 *Id.*, point 57 : « Toutefois, si, dans le cadre de la radiodiffusion télévisuelle d'une émission, une interface utilisateur graphique est affichée, les téléspectateurs reçoivent la communication de cette interface utilisateur graphique uniquement de manière passive, sans possibilité d'intervenir. Ils ne peuvent pas utiliser la fonction de ladite interface, qui consiste à permettre une interaction entre le programme d'ordinateur et l'utilisateur. ».

228 *Id.*, point 40 : « l'interface utilisateur graphique est une interface d'interaction, qui permet une communication entre le programme d'ordinateur et l'utilisateur ».

229 CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. c/ world programming Ltd*, points 39-41.

Une telle interprétation permet de répondre aux exigences de la doctrine opposée à la protection des logiciels par le droit d'auteur, car c'est alors la sensibilité se dégageant de l'œuvre qu'est l'interface graphique qui est protégée. Ce ressenti de l'utilisateur se retrouve dans le *game play* et les règles d'un jeu, « les modalités de l'interactivité »²³⁰, il se retrouve aussi dans les règles d'utilisation d'un langage de programmation.

L'arrêt du 2 mai 2012 de la Cour de justice de l'union européenne nous précise que les langages de programmation pourraient bénéficier en tant qu'œuvre de l'esprit de la protection du droit commun du droit d'auteur. Mais est-ce l'œuvre composite ou son élément central qu'est la règle qui est ainsi visée ?²³¹ Tout dépend de l'interprétation que l'on tire de l'arrêt *Bezpečnostní softwarová asociace*²³², mais par cohérence avec l'arrêt *Cryo* mieux vaut considérer qu'un nouvel objet est visé : la réglementation au cœur du langage de programmation.

76. – Outre ce point de vue théorique, il reste qu'en pratique c'est le régime distributif qui s'applique, la recherche d'unité s'impose.

B) Des mécanismes intéressants pour l'unité à atteindre

77. – Le régime unitaire est plus intéressant que le régime distributif (1), si en principe c'est le second qui s'applique, l'unité peut être atteinte en mettant en œuvre divers mécanismes juridiques (2).

1. L'intérêt mitigé de l'unité face à l'inacceptable éclatement

78. – Si l'unité du régime applicable à un objet doit être recherchée c'est d'abord dans un objectif évident de simplification, c'est ensuite pour sécuriser la gestion de la titularité, de l'exercice des droits et des exceptions. C'est avant tout le droit commun des programmes d'ordinateur que l'on va chercher à atteindre pour permettre une gestion simplifiée et sécurisée des droits, mais contrairement aux *a priori* ce régime est bien moins ouvert que le droit commun, étant spécial et ne devant pas bénéficier d'une condition d'originalité assouplie.

79. – **Gestion simplifiée des droits et éviction de l'auteur.** À l'inverse des investisseurs et producteurs, le salarié cherchera à maximiser la couverture du droit commun du droit d'auteur qui lui confère l'intégralité des droits d'auteur, toute cession devra être précise et spéciale²³³. Le

230 CSPLA avis n° 2005-1, 7 déc. 2005.

231 Cf. *supra*, n° 73-75.

232 CJUE, 22 déc. 2010, C-393/09, *Bezpečnostní softwarová asociace*.

233 L. 131-3 Alinéa 1 du Code de la propriété intellectuelle : « La transmission des droits de l'auteur est subordonnée à la condition que chacun des droits cédés fasse l'objet d'une mention distincte dans l'acte de cession et que le domaine d'exploitation des droits cédés soit délimité quant à son étendue et à sa destination,

cessionnaire et l'utilisateur devront respecter l'intégralité du droit moral, à savoir – outre le droit de divulgation, lequel permet en théorie au salarié de choisir de ne pas livrer l'œuvre créée, et le droit de paternité qui perdurent dans le régime spécial – d'une part le droit de retirer l'œuvre par repentance²³⁴, d'autre part le droit au respect de l'œuvre et de son intégrité, ce qui prohibe notamment le fait d'augmenter, modifier ou diminuer l'œuvre de façon à la dénaturer. Ce dernier est considérablement réduit dans le régime spécial, car un logiciel a vocation d'évoluer, ce droit se limite alors aux hypothèses où les modifications porteraient atteinte à la notoriété ou à l'honneur du créateur.

Le régime du droit commun devra être recherché par les auteurs pour contrôler l'exploitation de leur création, en particulier les partisans du libre désireux de diffuser gratuitement et largement le langage de programmation.

L'employeur bénéficiera d'un transfert *ab initio* des droits sur la forme programmée par son salarié dans le cadre de ses fonctions, il cherchera donc à étendre la notion de forme programmée aux langages de programmation. Pour l'utilisateur le bilan serait plutôt positif, il perd l'exception de copie privée remplacée par une simple copie de sauvegarde, mais obtient une exception de décompilation à des fins d'interopérabilité ainsi qu'un droit d'étudier le fonctionnement du logiciel afin de déterminer les idées à la base de chacun de ses composants. Plus qu'une exception il s'agit d'un droit, ainsi non seulement le titulaire des droits ne pourra interdire cette étude²³⁵, mais il ne pourra techniquement l'empêcher en se prévalant de la prohibition du contournement des mesures techniques. La Cour de justice, dans l'arrêt SAS, va jusqu'à admettre le droit de créer un logiciel concurrent à celui qui étudie un logiciel.

80. – Si une pluralité de régimes s'appliquent à la même œuvre, il s'agit de distinguer les apports couverts par le droit commun et ceux couverts par le droit spécial, sachant qu'ils peuvent émaner du même contributeur. L'utilisateur ne pourra employer certaines exceptions que pour une partie de l'œuvre seulement, *idem* concernant le droit moral du créateur ou de la titularité du producteur. On notera enfin qu'en matière de contrefaçon la bonne foi ne joue pas²³⁶, cela place le cessionnaire ou

quant au lieu et quant à la durée. »

234 À tout moment l'auteur peut choisir de retirer l'œuvre de la circulation, soit temporairement notamment pour l'améliorer, soit définitivement, ce pour des motifs d'ordre intellectuel

235 La mesure est d'ordre public européen selon : CJUE, 2 mai 2012, n° C-406/10, *SAS Institute Inc. c/ world programming Ltd*, point 53 : « L'article 9, paragraphe 1, de la directive 91/250 ajoute, par ailleurs, que toute disposition contractuelle contraire aux exceptions prévues à l'article 5, paragraphes 2 et 3, de cette directive sera nulle et non avenue. »

236 Depuis Cass. civ. 1^{re}, 6 juin 1990, *Textiles Maurice et autres c. SA Goutarel* ; sauf en matière pénale ou il n'y a qu'une présomption fragile de mauvaise foi selon l'interprétation des articles L. 335-2 et L. 335-3 du Code de la propriété intellectuelle faite par : LUCAS (A.), LUCAS (H.J.) et LUCAS-SHLOETER (A.), *Traité de la propriété littéraire et artistique*, 4e édition, 2012, LexisNexis, n° 1081.

l'utilisateur dans une position excessivement précaire. En revanche, est-il pertinent de rechercher la qualification de logiciel afin de bénéficier d'une condition d'originalité allégée ?

81. – **Une condition d'originalité prétendument plus souple.** Longtemps le droit spécial applicable aux programmes d'ordinateurs nous aura été présenté comme plus souple que le droit commun du droit d'auteur. La Cour de justice contredit l'analyse puisque son arrêt *Football Dataco* de mars 2012 uniformise le critère d'originalité et les arrêts *Bezpečnostní softwarová asociace* de décembre 2010 et *SAS contre WPL* de mai 2012 restreignent l'application du droit spécial au profit du droit commun.

82. – Comme le remarque très justement monsieur Roussel, l'œuvre, qu'il s'agisse d'un logiciel ou d'un langage de programmation, est soumise à la condition d'originalité indépendamment de son genre²³⁷, tout comme l'admettait déjà l'arrêt fondateur de la protection des logiciels par le droit d'auteur²³⁸. Toutefois, le rapport Jonquères qui a suivi ledit arrêt a soufflé un doute qui perdure encore et qui conduit certaines juridictions à rechercher l'originalité dans la nouveauté d'un programme ou l'activité inventive du programmeur. Encore récemment, le 17 octobre 2012²³⁹, la Cour de cassation sanctionnait pour insuffisance de motivation un arrêt d'appel qui se contentait de proclamer l'originalité d'un programme sans procéder à une démonstration de son existence.

Les juges du fond ont effectivement une approche sensiblement différente, voire sensiblement incohérente, il faut admettre que rechercher l'empreinte de la personnalité dans des lignes de code informatique n'est pas chose aisée. Pour une doctrine majoritaire qui le déplore²⁴⁰, un droit d'auteur à deux vitesses s'est développé diminuant l'exigence d'originalité en matière de programme d'ordinateur et se contentant d'un « effort personnalisé allant au-delà de la simple mise en œuvre d'une logique automatique et contraignante »²⁴¹, réservant aux œuvres d'art le critère de « l'empreinte de la personnalité de l'auteur », célèbre formule de Desbois.

Non seulement la jurisprudence originelle ne prévoyait pas la dualité du critère d'originalité, mais la récente jurisprudence de la Cour de justice de l'union européenne la condamne. Les arrêts *Painer*, *Infopaq*²⁴² et plus récemment l'arrêt *Football Dataco* du 1^{er} mars 2012, dans lequel l'originalité

237 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *CCE*, n° 4, avril 2005, étude 15, n° 4, §2.

238 Cass. AP, 7 mars 1986, *Pachot* ; *RTD Com.*, 1986, p. 399, obs. FRANÇON (A.) ; *RID Admin.*, juill. 1986, p. 136, obs. LUCAS (A.).

239 Cass. Civ. 1^{er}, 17 oct. 2012.

240 Cf. notamment POLLAUD-DULLIAN (F.), *Droit d'auteur*, Economica . Corpus, 2005 n° 118, 134 et 221.

241 Cass. AP, 7 mars 1986, *Pachot*, *loc. cit.*

242 CJUE, 16 juillet 2009, C-5/08, *Infopaq International* ; CCE 2009, comm. 97, note CARON (C.) ; JCP G, 2009, 272, note MARION (L.) ; *Propr. Intell.*, 2009, n° 33, p. 379, note BENABOU (V.-L.).

d'une base de données était en cause, tendent à confirmer la vision personnaliste du droit d'auteur à l'échelle européenne. Ce dernier arrêt précise qu'il faut rechercher la « touche personnelle » de l'auteur quelle que soit l'œuvre en cause et qu'il s'agit du seul critère valable, vis-à-vis des exigences européennes, pour apprécier l'originalité au sens du droit d'auteur. Ainsi le droit interne ne pourra s'en écarter. L'assemblée plénière l'écrivait en 1986, la Cour de justice le confirmait en 2012, l'originalité est la « touche personnelle », ou l'« effort personnalisé » de l'auteur, peu importe le type d'œuvre concernée. Rappelons que les langages de programmation peuvent être « exotiques, ésotériques, délirants, hilarants »²⁴³.

83. – Une fois l'exigence d'originalité uniformisée, la Cour de justice érige le droit d'auteur issu de la directive de 2001 en droit commun et celui issue de la directive de 1991 et de 2009, concernant les programmes d'ordinateur, en droit spécial. Ce n'est plus le droit spécial qui est plus souple mais le droit commun, car le premier ne protège que la forme qui est l'expression du programme²⁴⁴, c'est-à-dire les seules lignes de code programmées. Le rendu final qu'est l'interface graphique ferait l'objet d'un droit propre²⁴⁵. Il en est de même pour la création mise en œuvre par logiciel telle que le langage de programmation²⁴⁶.

84. – Ainsi, ce ne sera pas pour une originalité allégée que le droit applicable aux programmes d'ordinateur doit être recherché dans des montages juridiques, mais pour son régime.

2. *Des montages juridiques plutôt envisageables*

85. – Comme pour les œuvres multimédia il faut essayer de rattacher les éléments accessoires des langages à la qualification de programme avant de s'en remettre aux palliatifs attrayants au processus de conception et au contrat.

86. – **Application de l'*accessorium sequitur principale* à la syntaxe et aux néologismes.** À la différence des œuvres multimédia, les langages de programmation ne sont pas constitués, en principe, ni d'éléments graphiques ni d'éléments musicaux ou sonores et encore moins d'éléments audiovisuels, ce qui facilitera l'opération de « jonglerie » entre les qualifications. Les langages seraient des œuvres préparatoires à la conception de logiciels, ou bien ils seraient eux-mêmes des logiciels. L'analyse n'a pas été admise ni même rejetée par la Cour de justice, qui en avait pourtant

243 HUNAUT (G), [petite] *histoire des langages de programmation* [en ligne], <http://www.info.univ-angers.fr/~gh/hilapr/detour.htm>, consulté le 28 mai 2013.

244 CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. c/ world programming Ltd*, point 39 ; CJUE, 22 déc. 2010, C-393/09, *Bezpenostní softwarová asociace*, points 34 et 41.

245 Cf. *supra*, n° 75.

246 Cf. *infra*, n° 97-98.

l'occasion, elle rend une solution floue. Dans l'hypothèse où notre analyse devrait être validée, il faut s'entendre sur les éléments du langage touchés par la qualification de programme, les seules suites d'instructions ou bien faut-il incorporer les éléments accessoires qui les mettent en œuvre dans le langage de programmation : le choix des mots et néologismes ainsi que la syntaxe ? La syntaxe doit être rattachée au régime comme formant une entité inséparable des séries d'instructions, il en est de même des néologismes qui pourraient être assimilés à des titres d'œuvres.

87. – Concernant la syntaxe, il ne nous semble pas pertinent, bien que nous utilisions l'argument pour promouvoir sa protection comme élément au cœur du langage²⁴⁷, de considérer que les juges de Luxembourg souhaitent une application impérative du droit commun du droit d'auteur. Les séries d'instructions sont élaborées en résonance les unes par rapport aux autres, et la syntaxe permet la résonance et la cohérence. Elle établit les règles permettant aux séries binaires de se suivre de manière intelligible pour la machine. Plus qu'un accessoire indissociable de la série d'instructions, elle connaît avec elle une relation fusionnelle allant au-delà de l'accessoire. Sans cette syntaxe les différentes séries d'instructions n'auraient pas de raison d'être. Grâce à elle un langage est cohérent, on peut alors le décrire comme une série de séries d'instructions, à défaut on parlerait d'une base de séries d'instructions.

88. – Concernant les néologismes, il faut se référer au régime des titres des œuvres de l'article L. 112-4 du Code de la propriété intellectuelle. Il paraîtrait aberrant qu'un auteur qui aurait consenti une licence d'exploitation pour un ouvrage, attaque l'éditeur en contrefaçon de son titre par reproduction sans autorisation sur la couverture du livre. C'est donc bien que la licence couvre de manière implicite le titre du livre, ou bien que le titre suive le régime de l'élément principal. Bien entendu si un néologisme original préexistait et que sans autorisation l'auteur l'employait, il risquerait de subir une action en contrefaçon. Comme le précise l'article L112-4, le titre d'une œuvre est protégé « comme l'œuvre elle-même », il y aurait donc une relation d'accessoire qui a déjà été évoqué²⁴⁸. Cette relation d'accessoire a trait au régime applicable au titre, ce qui justifie la solution rendue dans l'affaire *Karbone 14* précédemment décrite²⁴⁹. Si l'œuvre principale n'est pas protégée, son titre ne l'est pas, à moins qu'il s'agisse d'un titre absolument original, hypothèse rarissime, qui serait alors protégé par le droit commun du droit d'auteur²⁵⁰. En étendant la solution à notre sujet, l'œuvre principale est un programme, le titre qui serait relativement original sera donc

247 Cf. *supra*, n° 58.

248 Cf. *supra*, n° 40-41 ; DESBOIS (H.), note sous Trib. Comm. Seine, 26 juin 1951, *Les hauts de Hurlevent*, RTD Com. 1952, p. 763, §7 : « Sa mission est de servir de pavillon à une œuvre de l'esprit, comme la marque à un produit de l'industrie : il importe de le protéger, non en lui-même, mais comme accessoire de l'œuvre. ».

249 *Idem*.

250 *Id.*

soumis au droit spécial du droit d'auteur « comme l'œuvre elle-même », le titre absolument original, au droit commun.

89. – Le recours à la qualification d'étape préparatoire dans la conception d'un programme d'ordinateur facilite notre opération. La syntaxe comme le lexique et les séries d'instructions qui s'y rattachent sont des éléments du tout qui permet la préparation de l'écriture du logiciel, comme le serait l'écriture de modules. Peu importe qu'il s'agisse ou non d'éléments programmés, ils bénéficieraient d'une qualification expresse de logiciel par assimilation, selon le treizième de l'article L. 112-1²⁵¹.

90. – **Processus de conception et voie contractuelle.** Le processus de conception et la voie contractuelle ne changent rien à la qualification juridique des éléments, seulement à la titularité des droits. L'existence d'un contrat quel qu'il soit avec le créateur du langage est indifférent, sauf pour les formes programmées, il faut prévoir spécialement chaque cession des droits en vertu de l'article L. 131-3 du Code de la propriété intellectuelle, toute stipulation étant interprétée *in favorem auctoris*²⁵². Les langages sont rarement le fait d'une personne unique, la plupart d'entre eux furent créés par des communautés scientifiques dans laquelle chaque acteur peut participer aux choix stratégiques ou le fait de société qui dirigera les contributeurs. Ce fut le cas du Fortran commandé par IBM. Seuls seront considérés comme auteurs ceux parmi les contributeurs qui auront effectivement réalisé un apport original à l'œuvre, ce qui supposera une analyse au cas par cas.

91. – La distinction entre œuvre collective et œuvre de collaboration est majeure et pourtant elle n'est pas aisée. Du point de vue de la titularité des droits patrimoniaux, tandis que « l'œuvre de collaboration est la propriété commune des coauteurs »²⁵³, « l'œuvre collective est, sauf preuve contraire, la propriété de la personne physique ou morale sous le nom de laquelle elle est divulguée »²⁵⁴. Pour résumer la distinction, dans l'œuvre de collaboration les contributeurs seront à une même échelle décisionnelle, de sorte que leur propre contribution elle-même originale participera de l'originalité de l'ensemble et qu'ils bénéficieront d'un droit de propriété exercé par chacun en indivision. Cela peut poser des problèmes de gestion des droits qu'il faut contractuellement résoudre, puisque la concession d'une licence supposera l'accord de chacun. À défaut d'accord l'intervention du juge sera requise. En revanche, dans l'œuvre collective un directeur gèrera les contributions de chacun, ainsi sera-t-il maître de la composition de l'ensemble et fera bénéficier de la titularité des droits celui qui divulguera l'œuvre.

251 « Les logiciels, y compris le matériel de conception préparatoire », *cf. supra* n° 55-56.

252 Latin signifiant : dans le sens le plus favorable à l'auteur.

253 Article L. 113-3 Alinéa 1 du Code de la propriété intellectuelle.

254 Article L. 113-5 Alinéa 1 du Code de la propriété intellectuelle.

Ces deux mécanismes peuvent se cumuler, plusieurs contributions collectives pouvant être réunies dans une œuvre de collaboration, notamment pour un projet mené en concertation entre plusieurs laboratoires, ou l'inverse. Par ailleurs, ils sont limités, car les contributeurs resteront titulaires des droits sur leur contribution personnelle pour une utilisation non concurrente à l'œuvre finale, sauf à prévoir spécialement et contractuellement la cession de ces droits. À défaut, la création et la distribution d'un langage différent serait envisageable s'il ne risque pas d'entraver la carrière du premier, notamment en le concurrençant.

92. – Ainsi, par principe, le droit commun et le droit spécial du droit d'auteur vont s'appliquer alternativement concernant les langages de programmation, la recherche d'un régime commun suppose d'importantes démonstrations, au moins pour identifier le titulaire des droits. Il reste que le ou les titulaires vont être amenés à exploiter leur langage de programmation, les formes sont diverses et des limites au droit d'auteur s'imposent.

II) Une exploitation multiforme et raisonnée des langages de programmation

93. – Les langages de programmation sont le monopole d'exploitation de leur auteur, ainsi pourrait-il librement déterminer les modes d'exploitation et le modèle économique qu'il désire (A), cependant, ses prérogatives sont spécialement limitées en droit d'auteur (B).

A) Les modes et modèles d'exploitation : le libre choix de l'auteur

94. – Sous réserve de respecter les droits d'auteur sur les œuvres antérieures, l'auteur va classiquement pouvoir choisir les modes d'exploitation de son œuvre (1), ainsi que le modèle économique de distribution entre le privatif et le libre (2).

1. Les trois modes d'exploitation incontournables

95. – L'auteur peut décider des conditions d'exploitation, mais un langage de programmation connaît trois utilisations naturelle, qui s'imposent d'elles-mêmes : permettre de programmer des logiciels, œuvres dérivées ; être mis en œuvre dans un instrument de programmation ; évoluer à travers l'apparition de nouveaux langages.

96. – **Le logiciel : une œuvre dérivée du langage.** Les langages de programmation sont des outils permettant la conception de programmes, le code objet et le code source du programme est donc entièrement imprégné du langage qu'il emploie. En effet, le lexique est directement reproduit

et la syntaxe respectée, la traduction en séries d'instructions est opérée de manière automatique par référence à celles écrites par le concepteur du langage. L'œuvre finale est donc une œuvre dérivée d'un langage, sous réserve que ce dernier soit une œuvre, sinon une œuvre composite par assemblage des quelques modules originaux. Tout programmeur qui utilisera un langage devra obtenir une licence d'utilisation, laquelle devra inclure le droit de reproduction voire le droit de distribuer une œuvre dérivée. C'est ici la clef de tout le danger, car l'utilisateur du programme litigieux se rendrait lui-même contrefacteur si le programmeur n'avait pas obtenu une telle licence. En effet, nul ne peut céder plus de droits qu'il n'en a. Le programmeur contrefacteur ne saurait autoriser un tiers à utiliser, par l'intermédiaire du programme final, le langage de programmation dans lequel il est écrit. Rappelons qu'en matière de contrefaçon la bonne foi ne joue pas²⁵⁵, l'utilisateur pourrait être condamné alors même que l'acheteur de bonne foi d'un bien volé serait remboursé. Une telle solution serait inacceptable, car l'utilisateur légitime n'est jamais en mesure d'apprécier s'il y a ou non contrefaçon, notamment au regard des logiciels libres.

Le langage est utilisé dans la phase d'écriture du programme, par l'assemblage des différents éléments, ainsi que dans la phase de compilation du code source et de décompilation du code objet par un processus de traduction automatique mis en œuvre par un instrument de programmation. Une exception de décompilation joue à l'égard des droits d'auteur sur le programme, mais joue-t-elle aussi à l'égard du langage ? La faculté d'interagir avec un programme écrit dans un langage est-elle soumise au droit d'auteur ? Il existe des moyens de limiter le monopole, il s'agira de l'étudier²⁵⁶.

97. – Les instruments de programmation et environnements d'exécution : œuvres composites, dérivées, collective ou de collaboration. « Sans les instruments de programmation qui autorisent sa mise en œuvre, un langage de programmation demeure au stade de l'exercice intellectuel »²⁵⁷. Il s'agit donc des logiciels qui permettent soit d'assister le concepteur pendant la programmation, soit de traduire automatiquement le code source en code objet, soit d'offrir un environnement d'exécution au langage. Ces différents instruments ne sont pas véritablement des œuvres dérivées du langage, celui-ci est intégralement reproduit sans modification dans un programme, il s'agit plutôt d'œuvres composites. D'autant plus que l'instrument et le langage sont souvent conçus concomitamment.

255 Depuis Cass. civ. 1^{re}, 6 juin 1990, *Textiles Maurice et autres c. SA Goutarel* ; sauf en matière pénale ou il n'y a qu'une présomption fragile de mauvaise foi selon l'interprétation des articles L. 335-2 et L. 335-3 du Code de la propriété intellectuelle faite par : LUCAS (A.), LUCAS (H.J.) et LUCAS-SHLOETER (A.), *Traité de la propriété littéraire et artistique*, 4e édition, 2012, LexisNexis, n° 1081.

256 Cf. *infra*, n° 106 et s.

257 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *op. cit.*, n° 1, §4.

Il faut aussi évoquer le cas des environnements d'exécution, logiciels indispensables à l'exploitation des programmes écrits dans un langage déterminé. Par exemple, les logiciels conçus dans un langage ne fonctionnant que sur *Windows* ne fonctionneront pas sur *Iphone*, sauf à passer par un émulateur ou langage intermédiaire comme le *JAVA*²⁵⁸. L'interface permettant l'exécution du code source comprend forcément les suites d'instructions écrites par le concepteur du langage, il les reproduit afin de pouvoir les interpréter et les exécuter. À l'instar des instruments de programmation, il s'agit donc d'œuvres composites mettant en œuvres les langages.

Une partie de la doctrine y a vu un moyen alternatif de protéger les langages de programmation²⁵⁹, mais il ne faut pas s'y tromper le langage et l'instrument de programmation sont deux objets bien distincts²⁶⁰, comme nous invite à le penser la solution rendue par la Cour de justice dans son arrêt *SAS contre WPL* du 2 mai 2012²⁶¹. « On voit immédiatement l'intérêt qu'il y aurait de considérer l'instrument de programmation comme une adaptation d'une œuvre préalablement existante, le langage de programmation »²⁶², comme le remarque monsieur Roussel, la mise en circulation de l'instrument de programmation supposera l'autorisation de son concepteur ainsi que celui du langage qui bien souvent est une même personne. S'agit-il d'une œuvre collective ou d'une œuvre de collaboration ? Le créateur du langage ne serait pas coauteur de l'instrument, d'après monsieur Roussel²⁶³, car cette « œuvre composite est la propriété de l'auteur qui l'a réalisée, sous réserve des droits de l'auteur de l'œuvre préexistante »²⁶⁴. Une analyse par analogie avec les œuvres multimédias nous amènera à une solution contraire, les graphismes sont mis en œuvre par le logiciel, ils sont bien entendus indépendant du logiciel ce qui n'empêche pas leur concepteur d'être coauteur de l'ensemble.

L'auteur ajoute qu'il sera difficile d'identifier une contrefaçon du langage par l'instrument, du fait d'une différence de nature et de seuils d'originalité, le premier étant une œuvre à caractère scientifique²⁶⁵. Toutefois, nous avons établi que le langage pourrait être lui-même une forme

258 *Cf. supra*, n° 28-29.

259 SANTOPAULO (M.-A.), *Protection juridique des langages de programmation*, Mémoire DEA, Montpellier, 2002, p. 22-26 ; ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *op. cit.*, n° 5-6 ; se contentant de souligner leur existence : SIRINELLI (P.), note sous Cour de Justice de l'Union Européenne (CJUE), grande Chambre, 2 mai 2012, affaire numéro C-406/10, *SAS Institute Inc. contre World Programming Ltd*, *RIDA*, juil. 2012, p. 221.

260 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *op. cit.*, n° 6, §4-6.

261 CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. c/ world programming Ltd.*, point 39 : « ni la fonctionnalité d'un programme d'ordinateur ni le langage de programmation et le format de fichiers de données utilisés dans le cadre d'un programme d'ordinateur pour exploiter certaines de ses fonctions ne constituent une forme d'expression de ce programme ».

262 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *op. cit.*, n° 6, §2.

263 *Idem*, n° 6, §4-6.

264 Article L. 113-4 du Code de la propriété intellectuelle.

265 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *op. cit.*, n° 6, §3.

programmée, qu'il a bel et bien une consistance. Il est forcément reproduit dans l'instrument, tout comme le français et l'anglais seraient forcément reproduits dans un logiciel de traduction.

98. – Enfin, un langage de programmation est une interface de programmation intermédiaire entre le code binaire et le langage humain, de façon à symboliser et synthétiser les commandes, nous l'avons dit, mais alors comment analyser un instrument de programmation offrant une interface purement graphique facilitant considérablement la programmation ? Plutôt qu'utiliser un traitement de texte l'utilisateur va recourir à des objets qu'il va pouvoir déplacer avec sa souris. On est alors à l'extrême limite de la programmation objet pour entrer dans le WSIWYG²⁶⁶, système surtout employé pour la création de sites web, pour les animations tridimensionnelles et la conception de jeux-vidéos. Il s'agit d'une nouvelle conception du langage de programmation plus proche de la compréhension humaine que le langage humain lui-même, par exemple si l'on souhaite que tel élément soit situé à tel endroit il suffit de le prendre (clic droit maintenu) et de l'y déplacer, un ensemble d'icônes permettra d'accéder aux différentes commandes disponibles. Chaque icône est à la fois un symbole potentiellement protégé par le droit d'auteur, permettant de désigner une commande écrite dans un langage de programmation, la syntaxe est entièrement assurée par le logiciel. La comparaison entre l'instrument de programmation et un logiciel de tampons prend alors tout son sens²⁶⁷. On ne peut nier qu'un tel programme, ou langage, soit une œuvre dérivée du langage dont il permet l'utilisation.

99. – **Le langage : une œuvre plus évolutive que le programme.** Les langages de programmation sont extrêmement nombreux, monsieur Kinnersley en liste 2500 dont 13 particulièrement influents²⁶⁸. Une véritable arborescence darwinienne se dessine entre les langages les plus importants et les autres²⁶⁹, amplifiant davantage le problème des œuvres dérivées qui vient d'être évoqué à l'égard des programmes²⁷⁰. Il y a bien œuvre dérivée dans le fait de modifier une œuvre, ce qui requiert l'autorisation de l'auteur, encore que l'on puisse assimiler une telle modification à une maintenance évolutive, donc à l'exception prévue à l'article 5.1 de la directive 91/250 à l'égard des programmes d'ordinateur réputés œuvres évolutives²⁷¹.

100. – Il ne sera pas toujours aisé de détecter la contrefaçon du langage dans un autre langage.

266 Acronyme « What you see is what you get » signifiant que vous obtenez le programme tel que vous le voyez à l'écran lorsque vous le concevez (traduit par nous).

267 Cf. *supra*, n° 50.

268 KINNERLEY (B.), *The language list*, [en ligne], <http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm>, consulté le 22 mai 2013.

269 Cf. LÉVÉNEZ (É.), *Computer languages history*, [en ligne], <http://www.levenez.com/lang/>, consulté le 22 mai 2013.

270 Cf. *supra*, n° 97-98.

271 Cf. *infra*, n° 127.

Monsieur Bigot nous éclair sur ce point²⁷² : d'une part parce que « les programmeurs utilisent de plus en plus des codes réutilisables libres de droits, des bibliothèques de classes en technologie objet par exemple, ou des logiciels Open Source » d'autre part parce que « les "ateliers de génie logiciel" fréquemment utilisés génèrent automatiquement des volumes importants de codes qui ne portent pas la marque de l'originalité d'un code écrit à la main », enfin parce que « certaines similitudes apparentes peuvent être imposées par le langage ou le contexte dans lequel il s'exécute, et ne pas relever d'une contrefaçon ». Pourtant, la recherche de la contrefaçon est rare en pratique puisque le modèle économique choisi est souvent le libre.

2. *Les modèles économiques : du rejet du privatif aux pièges du libre*

101. – **Modèle privatif et monopole conféré.** On ne peut dénier une sous-représentation du modèle privatif, qu'il s'agisse des langages ou des formats dits « propriétaires ». Le titulaire bénéficie pourtant d'un monopole d'exploitation complet, droit de reproduction et droit de représentation compris dans toutes leurs branches si le langage n'est pas un programme, sinon l'article L. 122-6 du Code de la propriété intellectuelle énonce les différents droits du monopole sur les logiciels. Par ailleurs, en vertu de l'article L. 131-3 du Code de la propriété intellectuelle les licences et actes de cession de droits d'auteur doivent d'une part être entendus strictement, d'autres part être précis quant-aux droits cédés, sous peine de nullité relative.

L'article L. 122-6 1 prévoit le droit d'autoriser « la reproduction permanente ou provisoire d'un logiciel en tout ou partie par tout moyen et sous toute forme », ce qui comprend évidemment les œuvres composites. L'article L. 122-6 2 prévoit le droit d'autoriser « la traduction, l'adaptation, l'arrangement ou toute autre modification d'un logiciel et la reproduction du logiciel en résultant », ce qui comprend les œuvres dérivées. Bien souvent le langage va être distribué avec un instrument de programmation, mais ce n'est pas un impératif, une licence d'exploitation pouvant être accordée à un développeur pour qu'il se charge d'une telle distribution, cela serait notamment envisageable pour le déploiement du langage sur de nouvelles plateformes. L'ayant droit pourra notamment autoriser ou interdire les différentes destinations du langage, exercer son droit de prêt et son droit de distribution. En principe il pourra interdire les éventuelles évolutions du langage.

102. – La seule prérogative qui pourrait être véritablement exclue du monopole d'exploitation d'un programme est le droit de représentation. Cette exclusion pourrait être confirmée par l'arrêt de la Cour de justice rendu le 3 juillet 2012²⁷³ qui appliquait la théorie de l'épuisement du droit de

272 BIGOT (J.-P.), « Détecter et matérialiser la contrefaçon d'un logiciel : la méthode américaine AFC Test, et l'outil logiciel de ESALAB », CCE n° 12, déc. 2006, étude 38, n° 1, §7.

273 CJUE, 2 mai 2012, C-406/10, SAS Institute Inc. c/ world programming Ltd.

distribution aux logiciels téléchargés en assimilant la licence d'exploitation à une cession. Or, le téléchargement se traduit par un acte d'émission et de réception de l'œuvre, soit un acte de représentation, quoi que l'on tienne l'exécution pour synonyme de la représentation. Il nous semble que l'acte de téléchargement consiste en réalité en une copie de l'œuvre par envois successifs des éléments copiés dans un emplacement distant, cet envoi n'est pas une émission mais une simple circulation de la copie dans la mesure où le programme n'est pas automatiquement exécuté contrairement à l'émission télévisuelle ou radiophonique. Un logiciel n'est véritablement représenté que lorsqu'il est exploité au moyen d'une machine mise à disposition du public ou lorsqu'il est exécuté en ligne, ce qui est le cas des programmes en flash : lecteurs de vidéo *streaming* et jeux exclusivement en ligne. Pour conforter cette analyse il faut se référer à l'arrêt de la Cour de justice rendu le 22 décembre 2010²⁷⁴, pour cet arrêt l'interface graphique d'un logiciel n'est pas représentée par une simple télédiffusion, sa représentation suppose une utilisation directe du logiciel par le public. Il faut d'ailleurs noter qu'un langage de programmation pourrait être jugé comme n'étant pas un programme mais une œuvre soumise au droit commun bénéficiant d'un droit de représentation, nous l'avons évoqué, or si le programme est représenté, le langage qui est nécessairement reproduit dans le programme l'est aussi.

103. – La difficulté de rémunération des créateurs de langages de programmation tient autant du grand nombre de participants à leur création, sans compter les diverses filiations entre langages, que du nombre de droits d'auteur mis en œuvre pour chaque programme. De ce fait les seuls modèles véritablement envisageables seraient : la gestion collective, voire une éventuelle licence globale, ou une licence libre.

104. – **Le choix d'un modèle libre.** Le libre est pratiquement synonyme de gratuité, or les articles L. 122-6 2 et L. 122-7-1 du Code de la propriété intellectuelle envisagent la mise à disposition à titre gratuit de l'œuvre, l'auteur à ce pouvoir quel que soit le droit susvisé qui serait exercé. Toutefois, le rapport Lescure opère une distinction entre les notions²⁷⁵ : on peut construire un modèle commercial autour d'œuvres dites « libres », la distribution pouvant être payante, et un modèle « gratuit » autour d'œuvres non-libres, le recours à la publicité et à l'impôt comme modes de financement de la télévision le démontrent. Pour autant la gratuité ou la liberté diffèrent de la mise à disposition inconditionnelle. D'abord, un écrit semble indispensable afin de respecter les conditions de l'article L. 131-3 du Code de la propriété intellectuelle, c'est-à-dire viser de manière

274 CJUE, 22 déc. 2010, C-393/09, *Bezpečnostní softwarová asociace*.

275 LESCURE (P.), *Mission culture – Acte 2 . Contribution aux politiques culturelles à l'ère numérique*, rapport au ministère de la culture [en ligne], mai 2013, p. 258 et 455, accessible sur : http://www.culturecommunication.gouv.fr/var/culture/storage/culture_mag/rapport_lescur/index.htm.

précise chacun des droits envisagés par la licence. Ensuite, un certain formalisme peut s'imposer, car l'article L. 131-2 exige un écrit à titre probatoire²⁷⁶ pour toute « autorisation gratuite d'exécution », or, nous l'avons vu, un droit de représentation pourrait être accordé en matière de langages.

Enfin, ce serait une erreur de permettre une utilisation sans condition de l'œuvre, notons que cela semble impossible au regard de l'article L. 131-3 du Code de la propriété intellectuelle. Pierre Lescure proposait pourtant que soit adoptée une définition positive du domaine public²⁷⁷ et qu'il soit possible de faire tomber l'œuvre par anticipation dans le domaine public. Ce serait envisageable, sauf à l'égard du droit moral contrairement à ce que proposaient les licences *Public Domain Dedication and Licence* et *CC zéro*²⁷⁸. Il nous semble que ces licences posent plus de problèmes qu'elles n'en résolvent. Premièrement, elles semblent incompatibles avec exercice du droit de retrait et de repentir. Deuxièmement, si l'auteur est mu par un libéralisme absolu il lui suffirait, à chaque demande d'exploitation exceptionnelle de ses droits, d'établir une nouvelle licence. Est-il besoin d'aller jusqu'à accorder une licence aussi absolue au risque que surgisse une exploitation nouvelle, non prévue et indésirable ? Par exemple, l'utilisation dans un cadre publicitaire, le détournement au détriment de l'auteur, un irrespect qui porterait atteinte à son honneur ou à sa notoriété. Même en matière de langages de programmation libres le secours du droit d'auteur est appréciable²⁷⁹. Pour s'en convaincre il suffit d'évoquer l'affaire *Sun Microsystem contre Microsoft*²⁸⁰ dans laquelle Microsoft avait modifié quelques mots à un langage de programmation dont il s'était fait concéder l'usage des codes sources de l'instrument de programmation afin de rendre ce logiciel « techniquement indissociable d'un logiciel tiers dont il était, par ailleurs, titulaire exclusif des droits y afférents »²⁸¹. Dans pareille hypothèse il ne resterait plus à l'auteur que la responsabilité civile²⁸². Un autre événement nous rappelle l'importance de conserver une marge de manœuvre dans les licences libres, ce fut le choix délibéré de *Microsoft* de ne pas respecter les normes internet W3C, après que le HTML eut été rendu libre d'utilisation, le géant américain contraignit pendant longtemps les *webmasters* à créer systématiquement deux versions de leurs pages internet ou de se contenter de celle de *Microsoft*. *Internet explorer*, qui disposait donc de ses propres interprétations

276 Cass. Civ. 1^{re}, 12 avr. 1976, Bull. civ. I, n° 123.

277 LESCURE (P.), *Mission culture – Acte 2 . Contribution aux politiques culturelles à l'ère numérique*, op. cit., p. 496, propositions 74-76.

278 *Idem*, p. 457.

279 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *CCE* n° 4, avr. 2005, étude 15, n° 7, §3 : « Il importe cependant de se garder des comportements prédateurs afin d'éviter, dans les faits, un dessaisissement de la chose mise à disposition. ».

280 Cour du district de San Jose, 9 sept. 1998, *Sun contre Microsoft*.

281 ROUSSEL (P.), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *loc. cit.*

282 *Ibidem*.

du langage HTML, était, au début des années 2000, le navigateur le plus utilisé, car systématiquement distribué sur les systèmes d'exploitation de loin les plus populaires de l'époque : *Windows* (98, 2000, ME et XP). Le droit d'auteur permet de condamner ce genre d'abus, sous réserve que l'on ne s'en sépare pas.

105. – Les licences libres sont très variables et se mélangent parfois avec le modèle propriétaire. Ainsi le rapport Lescure présente deux familles de licences libres : les *Creatives commons* et les licences de l'*Open knowledge foundation*²⁸³. Les variantes entre les différentes licences tiennent surtout à la teneur du droit moral que l'auteur souhaite conserver ou bien la faculté de se réserver le droit d'autoriser les exploitations commerciales, la distribution d'œuvres dérivées.

Le point le plus intéressant pour les partisans du libre est que leur choix va pouvoir influencer celui de l'utilisateur sur son exploitation des programmes – œuvres dérivées du langage – qu'il va créer, notamment par le biais de la licence CC-BY-NC-SA selon laquelle : « *Toute exploitation de l'œuvre est permise sous les mêmes conditions que CC-BY-ND, mais la création d'œuvres dérivées n'est autorisée qu'à la condition que ces œuvres soient distribuées sous une licence identique à celle qui régit l'œuvre originale.* »²⁸⁴.

106. – Ainsi le créateur du langage bénéficie d'un pouvoir extrêmement large, son autorisation spéciale étant requise pour toute exploitation nouvelle. Un pouvoir qu'il faut sagement exercer sans tomber dans l'excès de l'abandon pur et simple au risque de le regretter, sans tomber non plus dans l'excès d'un système qui rendrait « captif » l'utilisateur, au risque de se voir opposer une limite au droit d'auteur.

B) Le langage : un droit limité et une entrave aux exceptions

107. – Le droit d'auteur sur le langage pourrait être une entrave à la mise en œuvre des exceptions au droit d'auteur sur les programmes, mais il peut être excepté à ce droit. Pour exécuter un programme, comprendre son langage est indispensable, il faut disposer des traductions de chaque élément pris isolément. Dans l'affaire *SAS Institute contre WPL*²⁸⁵, la société créatrice du langage rendait les utilisateurs captifs de son système puisqu'aucun concurrent n'était autorisé à le lire. Cela est d'autant plus préjudiciable lorsque le langage devient obsolète. Or, deux solutions permettent de sortir de la captivité : traduire le logiciel dans un autre langage (1) ou recourir à

283 LESCURE (P.), *Mission culture – Acte 2 . Contribution aux politiques culturelles à l'ère numérique*, op. cit., p. 455-457.

284 *Idem*, p. 456.

285 CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. c/ world programming Ltd.*

l'interopérabilité (2).

1. *Droit de la traduction des programmes*

108. – Plusieurs acteurs peuvent s'intéresser à la traduction d'un programme, d'abord l'ayant droit sur le programme afin que celui-ci soit disponible dans un autre environnement d'exécution, l'utilisateur du programme pour pouvoir l'utiliser s'il ne dispose pas d'un environnement d'exécution adaptée, il s'agit alors d'œuvres dérivées. Enfin la libre concurrence permet à un concurrent de traduire lui-même le logiciel, ce qui constituera selon les circonstances une œuvre nouvelle ou une œuvre dérivée.

109. – **Droit à la traduction : une sortie de la captivité.** Le créateur et l'utilisateur d'un logiciel ont le droit de le traduire. En effet, le processus de création d'un programme consiste à l'écrire dans un langage à partir d'un organigramme. La mise à disposition sur une autre plateforme suppose simplement de reprendre l'organigramme et d'écrire à nouveau. L'utilisateur légitime d'un programme bénéficie d'une licence sur l'ensemble, c'est-à-dire le logiciel en lui-même et en tant qu'œuvre dérivée du langage. L'utilisateur comme l'ayant droit pourront donc licitement traduire le logiciel dans un autre langage. L'ayant droit sur le langage ne pourra s'opposer qu'au développement d'un logiciel qui procéderait automatiquement à la traduction, un tel logiciel serait un instrument de programmation qui intégrerait le langage, ce serait une œuvre dérivée. S'il souhaitait contractuellement interdire la traduction, il s'opposerait à un véritable droit à la traduction. Bien souvent l'ayant droit sur le langage et sur l'instrument de programmation seront une seule et même personne, si bien qu'il cherchera à empêcher l'émergence de logiciels susceptibles de lire son langage en revendiquant aussi bien ses droits sur son langage que sur son instrument.

110. – À l'égard de l'ayant droit sur le programme, l'utilisateur pourra faire valoir un droit à la traduction. Pour le démontrer il faut faire un parallèle avec le droit à la traduction française des textes en langue étrangère et avec la limitation du droit sur les mesures techniques de protection. Pour mettre en évidence un tel droit à la traduction au profit du consommateur, monsieur Relmy²⁸⁶ se réfère à l'article 2 de la loi n° 94-665 du 4 août 1994 lequel impose l'emploi de la langue française notamment dans les modes d'emploi ou d'utilisation. Pour la Cour d'appel de Paris, dans son arrêt du 10 février 2003, la traduction « était nécessaire pour l'acheteur du produit », nécessaire en effet pour pouvoir utiliser le produit, car il fallait au préalable savoir comment l'utiliser²⁸⁷. La

286 RELMY (J.-P.), *Le droit de la traduction . Contribution à l'étude du droit du langage*, thèse soutenue en 2007, Paris-Sud, n° 92 et s.

287 Comme le souligne RELMY (J.-P.), *idem*, n° 93.

traduction d'un logiciel peut elle-même être indispensable à son utilisation.

La faculté de traduire pourrait même échapper au droit de l'auteur, soit qu'il y a mise en œuvre de l'exception de copie privée, soit qu'un droit d'accès à la culture s'y oppose. D'abord, la traduction d'un logiciel dans un autre langage n'est-elle pas une copie privée de l'œuvre traduite dès lors qu'elle n'est destinée qu'à un usage personnel et familial ? Si l'exception de copie privée devrait être admise en matière de traductions de textes²⁸⁸, il ne pourra en être qu'autrement à l'égard des programmes pour lesquels seule l'exception de copie de sauvegarde sera envisageable. L'utilisateur ne pourra que revendiquer un droit à la maintenance adaptative de son logiciel et sous réserve que le changement de langage de programmation soit strictement imposé par l'évolution des machines. Ensuite, d'après madame Basalamah : « *Le droit de la traduction conduit au droit à la culture et à la diversité culturelle en ce qu'il appelle à une plus grande accessibilité aux produits culturels, à leur transformation et à leur régénération permanente.* »²⁸⁹. Un véritable contrat social entre l'auteur et l'humanité imposerait de rendre l'œuvre disponible²⁹⁰. L'auteur va très loin, voire trop loin, en considérant que ce n'est pas l'auteur de la traduction qui a une dette envers le premier auteur pour lui avoir emprunté son œuvre, mais l'inverse du fait de la survie de l'œuvre dans la traduction²⁹¹.

111. – Un parallèle avec les mesures techniques de protection permet d'envisager d'autres angles d'attaque. Pour monsieur Chen²⁹², il est possible d'écarter de telles mesures qui rendraient captif l'utilisateur en recourant au droit civil et au droit de la consommation. Le droit commun avec la garantie des vices cachés, notamment le vice que serait l'impossibilité d'exécuter le programme sur une plateforme, et avec la théorie de l'abus de droit d'auteur sur le langage ou sur le programme en prohibant la traduction. Ce dernier point permettrait peut-être d'outrepasser l'interdiction de créer un logiciel de traduction automatique. Le droit de la consommation plaide davantage en faveur de l'exception d'interopérabilité que du droit à la traduction²⁹³.

112. – Enfin, dans certaines conditions le concurrent lui-même peut traduire pour créer un logiciel de substitution, il en sera alors le seul titulaire.

288 ZÉRAOUI-SARAH (Farha) « *Les traductions : le régime de protection par le droit d'auteur . Étude comparative droit algérien-droit français* », Propriétés intellectuelles n° 46, janvier 2013, p. 35.

289 BASALAMAH (S.), *Le droit de traduire . Une politique culturelle pour la mondialisation*, Les Presses de l'Université d'Ottawa, 2009, p. 409.

290 *Idem*, p. 410.

291 *Id.*, p. 412.

292 CHEN (S.-T.), *Droit d'auteur et protection technique des œuvres – étude de droit comparé européen et asiatique*, thèse soutenue le 13 déc. 2010, Nantes, p. 314-341, n° 497-532.

293 *Cf. infra*, n° 119.

113. – **Le droit sur la traduction : une libre concurrence.** L'arrêt *Winsure*²⁹⁴, rendu par la première chambre civile de la cour de cassation en 2011, doit ici retenir l'attention en ce qu'il accorde à un concurrent : le droit de traduire un logiciel dans un autre langage de programmation, ainsi que l'intégralité des droits sur ce nouveau programme. En effet, ce dernier ne sera pas considéré comme étant une œuvre dérivée du premier. En l'espèce deux programmeurs avaient déposés, à fin de preuve de leur création, en 2000 un logiciel appelé *winsure II STG*. En 2001 l'un d'eux créa un logiciel similaire, *CMT*, dans un autre langage puis le dépose ; l'autre reconfigura *Winsure II STG* dans deux nouvelles versions, *VAI* et *ADT*, sans changer de langage de programmation. Chacun attaquant l'autre en contrefaçon, la Cour d'appel considéra que la version concurrente créée dans un nouveau langage n'était pas une œuvre dérivée « du seul fait que [le nouveau logiciel] constituait un nouveau développement dans le langage *Windev* »²⁹⁵. En revanche l'autre comportement, qui correspondait simplement à reconfigurer le logiciel, fut condamné comme étant contrefaisant en ce qu'il consistait à créer des œuvres dérivées sans autorisation. La Cour de cassation confirme l'arrêt d'appel, une traduction d'un logiciel n'est pas une contrefaçon, mais un nouveau logiciel, car créé dans un autre langage « selon des méthodes distinctes de celles utilisées pour les logiciels précédents »²⁹⁶ qu'il imite, ce qui suffit à caractériser son originalité²⁹⁷.

114. – Bien que non publiée au bulletin, cette solution est notable, elle confirme celle rendue par la Cour d'appel de Rennes en 2006²⁹⁸, selon laquelle il y a contrefaçon dès lors que le contrefacteur ne peut démontrer soit que les similitudes avec le logiciel imité ne sont que résiduelles, soit qu'il l'a créé dans un langage différent. La portée de cette dernière décision était jusqu'alors limitée puisqu'elle concernait une contrefaçon de marque et non pas de droit d'auteur. Il reste que la solution est cohérente puisque, nous l'avons dit, traduire suppose de reprendre l'élaboration du logiciel à l'étape de l'organigramme, or un organigramme n'est pas en soi considéré comme protégé par le droit d'auteur²⁹⁹. Madame Basalamah cherche à l'expliquer, la traduction et l'œuvre originale n'auraient pas un lien de filiation mais de fraternité³⁰⁰, il s'agirait de deux œuvres nouvelles distinctes l'une de l'autre bien qu'elles expriment les mêmes idées. Elle ajoute que : « *De fait, il n'y a pas plus d'originalité dans une "œuvre originale" que dans la mémoire extensive de l'humanité.*

294 Cass. civ. 1^{re}, 22 sept. 2011, n° 09-71.337 : *JurisData* n° 2011-019931 ; *RIDA* 2011 n° 230, p. 149 et p. 249, obs. SIRINELLI (P.) ; *RTD Com.* 2011, p. 741, obs. POLLAUD-DULIAN (F.).

295 *Ibidem*.

296 *Ib.*

297 Relevant cette précision pour la critiquer : SARDAIN (Frédéric), « Hors-piste n° 1 » in « Propriété littéraire et artistique », *La semaine juridique entreprise et affaires* n° 28, 12 juil. 2012, 1461.

298 CA Rennes 3^e ch. cor., 23 fév. 2006, n° 05/01416.

299 Sauf en ce qu'il pourrait constituer une étape préparatoire dans la création de logiciels.

300 BASALAMAH (S.), *Le droit de traduire . Une politique culturelle pour la mondialisation, op. cit.*, p. 412.

On ne crée jamais rien, on ne fait que renouveler son regard sur ce qui est. »³⁰¹. Or, il n’y a pas contrefaçon s’il n’y a reprises que des idées et des éléments qui ne sont pas spécialement le siège de l’originalité³⁰².

115. – Pourtant, cette jurisprudence est contestable. D’abord, l’article L. 122-6 2 du Code de la propriété intellectuelle vise expressément le droit pour le programmeur d’autoriser la traduction de son logiciel, or le moyen, « fort mal motivé »³⁰³, ne soulève que les articles L. 112-1 et L. 112-2. La notion de traduction ne correspond pas qu’à la traduction des textes compris dans le logiciel, mais aussi la traduction en différents langages de programmation³⁰⁴. Ensuite, le rejet même de la protection des organigrammes est contestable, le siège de l’originalité d’un programme ne réside pas que dans chacun de ses modules pris isolément, mais aussi dans la composition de l’ensemble, or c’est l’organigramme qui, à l’instar d’un synopsis, selon son degré d’élaboration, organise cette composition de l’ensemble. Un mouvement doctrinal américain avait, en vain, procédé par analogie avec les plans d’architecture³⁰⁵. On ne saurait revendiquer d’une part que l’originalité d’une œuvre puisse résider dans sa composition et d’autre part rejeter la protection du plan qui l’organise, sous réserve de pouvoir percevoir l’originalité d’un tel plan, ce qui suppose un degré de précision particulièrement avancé. À titre d’exemple, ce n’est que pour insuffisance d’originalité qu’un plan de cours de droit pénal n’a pas bénéficié de la protection du droit d’auteur³⁰⁶, c’est donc que s’il avait été original il aurait été protégé³⁰⁷.

116. – La traduction du logiciel est donc toujours possible et même encouragée, cela n’en représente pas moins un travail de longue haleine d’autant plus que cette limite ne permet pas de créer de logiciels concurrents dans le même langage de programmation. D’où l’intérêt d’établir un droit d’étudier le logiciel à des fins d’interopérabilité et de création de logiciels concurrents.

2. *Droit d’étudier le logiciel à des fins d’interopérabilité et de création*

117. – Face à l’insuffisance d’un droit à la traduction pour sortir de la captivité d’un langage ou d’un format propriétaire, il faut rechercher d’autres limites au droit d’auteur. L’utilisateur bénéficie de deux exceptions : le droit à l’interopérabilité et le droit d’étudier le fonctionnement du logiciel et

301 BASALAMAH (S.), *Le droit de traduire. Une politique culturelle pour la mondialisation*, op. cit., p. 413.

302 MACREZ (F.), « Le droit d’auteur, le programme d’ordinateur et la cour de justice », *RIDA*, 22 octobre 2012, note de bas de page n° 32.

303 SARDAIN (Frédéric), « Hors-piste n° 1 » in « Propriété littéraire et artistique », loc. cit.

304 POLLAUD-DULIAN (F.), *RTD Com.* 2011, p. 741, obs. sous Cass. civ. 1^{re}, 22 sept. 2011, n° 09-71.337.

305 D. SCOT (M.), *Computer Law*, Wiley law publications, 1984, §3.33.

306 Cass. crim., 18 oct. 2011, n° 11-81.404.

307 Critiquant la solution de la décision de 2011 au regard du droit d’auteur : BOULOC (B.) comm. de Cass. crim., 18 oct. 2011, n° 11-81.404, *RTD Com.* 2012 p. 202.

du langage.

118. – **Droit à l'interopérabilité avec le langage ou le format.** L'interopérabilité est la « capacité d'échanger des informations et d'utiliser mutuellement des informations échangées »³⁰⁸. La notion est à rapprocher de la compatibilité bien qu'il ne faille pas les confondre³⁰⁹. Il faut faire un parallèle entre l'emploi d'un langage de programmation propriétaire et l'emploi d'une mesure technique de protection, tous deux étant des logiciels qui ont la fâcheuse tendance à créer une exclusivité de fait sur des œuvres. Théoriquement les solutions dégagées à l'égard des mesures techniques de protections devraient s'appliquer en la matière, il reste encore à le vérifier en pratique.

Il n'y a pas véritablement d'exception d'interopérabilité³¹⁰, mais comme seul le code source d'un logiciel permet de le comprendre et de le modifier, s'il n'est pas fourni par son créateur il peut être obtenu par l'intervention d'une autorité ou par décompilation du logiciel. On distingue ainsi « l'interopérabilité "autoprocuree" » de « l'interopérabilité obtenue à l'aide d'une intervention publique »³¹¹.

119. – Il est d'abord possible de demander l'intervention d'une autorité publique soit par le consommateur pour vice caché, vente liée et tromperie, soit par un concurrent pour irrespect des règles de concurrence, enfin par quiconque pour abus de droit.

Le cas du vice caché et de l'abus de droit a déjà été vu³¹², en revanche la condamnation pour vente liée et tromperie constituent un point important de la réflexion de monsieur Chen³¹³. La société Sony avait été condamnée pour vente liée et tromperie³¹⁴ puisque ses mesures techniques de protection rendaient totalement indépendant son baladeur avec son service de vente en ligne de fichiers musicaux. Pour autant, on peut se questionner sur la valeur de cette prohibition des ventes liées face à la directive 2011/83/UE sur le droit de la consommation³¹⁵, puisque celle-ci interdit les mesures générales de prohibition en matière commerciale. Quant au délit de tromperie, il est

308 Considérant 12 de la Directive 91/250/CEE du conseil, 14 mai 1991, concernant la protection juridique des langages de programmation.

309 CHEN (S.-T.), *Droit d'auteur et protection technique des œuvres – étude de droit comparé européen et asiatique*, *op. cit.*, p. 314 n° 494.

310 SIRINELLI (P.), note sous CJUE, grande ch., 2 mai 2012, affaire numéro C-406/10, *SAS Institute Inc. contre World Programming Ltd, RIDA*, juil. 2012, p. 299.

311 CHEN (S.-T.), *Droit d'auteur et protection technique des œuvres – étude de droit comparé européen et asiatique*, *op. cit.*, p. 315-339, n° 496-530.

312 *Cf. supra* n° 111.

313 CHEN (S.-T.), *Droit d'auteur et protection technique des œuvres – étude de droit comparé européen et asiatique*, *op. cit.*, p. 331-332, n° 516-518

314 CHEN (S.-T.), *Droit d'auteur et protection technique des œuvres – étude de droit comparé européen et asiatique*, *op. cit.*, p. 331.

315 Directive 2011/83/UE, 25 oct. 2011, relative aux droits des consommateurs.

rarement constitué, les plateformes compatibles avec le logiciel étant généralement indiquées.

Les règles de la concurrence permettent le recours à la théorie des facilités essentielles, sous réserve que le langage de programmation en question bénéficie d'un véritable monopole, ce qui n'était pas le cas dans l'affaire *Apple iTunes* concernant l'incompatibilité de mesures techniques de protection entre deux concurrents³¹⁶. De par la généralité des termes de l'article L. 331-32 Alinéa 1 du Code de la propriété intellectuelle³¹⁷ il devrait être aussi possible de faire ordonner par la HADOPI la communication des moyens de l'interopérabilité avec les programmes écrits dans un langage propriétaire, mais une telle solution reste théorique.

120. – Il est ensuite possible, pour l'utilisateur légitime du logiciel, d'en traduire la forme exécutable en une forme compréhensible par l'homme par voie de décompilation. L'utilisateur est donc contraint d'utiliser le langage de programmation, il bénéficierait d'une sous-licence implicite d'utilisation du fait de sa licence sur le logiciel lui-même. Une telle décompilation n'a rien de parfait³¹⁸, d'autant plus que l'utilisateur ne bénéficiera pas des commentaires écrits par le programmeur entre les lignes de code, de tels commentaires permettent de mieux comprendre le programme. Dans la version écrite par le programmeur on pourrait trouver ce genre de commentaires : « le module suivant (ligne 320 à 372) permet le copier-coller ». La pression des lobbies a permis de limiter cette décompilation aux modules servant à l'interopérabilité³¹⁹, et elle ne pourra être mise en œuvre qu'en cas de refus du créateur de fournir les informations permettant l'interopérabilité. L'esprit de l'exception est donc de permettre la création de l'interopérabilité et pas seulement de se contenter d'obtenir des informations. Cela suppose de reproduire le langage de programmation, ou même les éléments logiciels permettant la lecture d'un langage de programmation.

121. – L'objectif d'interopérabilité justifie que l'on excepte au droit d'auteur sur le langage, sous réserve que celui-ci soit considéré comme un programme. L'exception ou le droit à l'interopérabilité devrait permettre la migration de données ou de logiciels d'une plateforme propriétaire à une autre. Cependant, dans l'affaire *Fiducial contre DPSI*³²⁰ n'étaient admises que les

316 Cons. Conc., 9 nov. 2004, déc. n° 04-D-54, *Apple iTunes contre VirginMega*.

317 « Tout éditeur de logiciel, tout fabricant de système technique et tout exploitant de service peut, en cas de refus d'accès aux informations essentielles à l'interopérabilité, demander à la Haute Autorité de garantir l'interopérabilité des systèmes et des services existants, dans le respect des droits des parties, et d'obtenir du titulaire des droits sur la mesure technique les informations essentielles à cette interopérabilité. A compter de sa saisine, la Haute Autorité dispose d'un délai de deux mois pour rendre sa décision. »

318 Le code est fourni en langage assembleur, donc dans un langage différent selon : MACREZ (F.), *Créations informatiques : bouleversement des droits de propriété intellectuelle ? Essai sur la cohérence des droits*, Litec . Collection du CEIPI, 2011, p. 377, n° 472.

319 *Idem*, n° 80.

320 Cass. civ. 1^{re}, 20 oct. 2011, n° 10-14.069, F-P+B+I, *Sté Fiducial c/ Sté Développement professionnel spécialisé informatique*.

opérations qui, « pour récupérer les fichiers de ce programme, s'inscrivaient dans les strictes nécessités de l'interopérabilité autorisée par l'article L. 122-6-1, IV » du Code de la propriété intellectuelle. Ainsi la cour procède à une « interprétation fantaisiste »³²¹ de l'exception de décompilation à des fins d'interopérabilité pour légitimer l'extraction des fichiers. L'exception de décompilation permet alors l'extraction et la migration de données écrites dans un format propriétaire, par extension il faut appliquer la solution aux programmes écrits dans un langage propriétaire. Dans l'affaire *SAS contre WPL*³²², un concurrent avait créé un logiciel destiné à lire les scripts, petits programmes, conçus par le biais d'un autre logiciel. Il y avait donc nécessairement interopérabilité entre le nouveau logiciel d'accueil et les scripts. Seulement, il ne semble pas que les créateurs des scripts puissent accorder une licence d'utilisation sur le langage qui ne leur appartient pas, leur propre licence étant très limitée, d'où la mise en œuvre d'une exception au droit d'auteur.

122. – On notera enfin que dans chacune des affaires il n'y avait pas eu décompilation du code source, mais simplement étude des *inputs* et des *outputs* afin de créer un logiciel de substitution, ce qui a été légitimé.

123. – **Étude du logiciel et du langage et création de forks**³²³. Dans l'affaire *Fiducial contre DPSI*³²⁴ comme dans l'affaire *SAS contre WPL*³²⁵, susvisées, les concepteurs imitateurs se sont contentés d'une étude du logiciel originel destinée à déterminer le fonctionnement de l'interface entre les données ou les scripts à faire migrer, d'une part, et le logiciel d'accueil originel, d'autre part, pour ensuite reproduire cette interface dans le logiciel d'accueil concurrent. Dans la seconde affaire, la Cour de justice reconnaît le caractère impératif de l'exception d'étude du fonctionnement d'un logiciel, mais on pourrait considérer qu'elle méconnaisse les droits d'auteur sur l'interface entre le logiciel d'accueil et les scripts en permettant ce que l'on peut considérer comme étant du « surmoulage ».

On ne reprochera pas à la cour d'avoir admis comme étant d'ordre public le droit d'étudier le fonctionnement d'un logiciel³²⁶, puisque la solution est de droit positif en France³²⁷. En matière de

321 SARDAIN (Frédéric), « Hors-piste n° 2 » in « Propriété littéraire et artistique », La semaine juridique entreprise et affaires n° 28, 12 juil. 2012, 1461.

322 CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. c/ world programming Ltd.*

323 De l'anglais « fourche » ou « embranchement », il s'agit d'un logiciel issu ou fortement inspiré d'un autre logiciel, bien souvent il en reprend une partie du code source.

324 Cass. civ. 1^{re}, 20 oct. 2011, n° 10-14.069, F-P+B+I, *Sté Fiducial c/ Sté Développement professionnel spécialisé informatique.*

325 CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. contre world programming Ltd.*

326 SIRINELLI (P.), note sous CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. contre World Programming Ltd, RIDA*, juil. 2012, p. 289 : « Ne pas admettre ce principe pourrait avoir des conséquences peu heureuses comme permettre, au profit du créateur du premier logiciel, une réservation indirecte, contractuelle, des idées et des principes à la base de tout élément d'un programme d'ordinateur et cela alors même que ces éléments sont réputés être de libre parcours. ».

327 Art. L. 122-6-1 5 al. 2 du Code de la propriété intellectuelle : « Toute stipulation contraire aux dispositions

langages de programmation, c'est à la suite du conflit précité³²⁸ entre Microsoft et Sun Microsystems, à propos de la modification par la première société du langage JAVA, qu'est né le langage C# aux propriétés similaires. Ainsi, une société ne peut modifier sans autorisation le langage créé par un tiers, mais est libre de créer un langage concurrent.

124. – Cette décision de la cour de justice reconnaît aussi l'existence de droits d'auteur sur le langage. En principe, si logiciel de substitution utilise sans autorisation le langage de programmation, il le reproduira nécessairement et il y aura contrefaçon du langage dans le logiciel de substitution, sauf à recourir à l'exception d'interopérabilité entre le logiciel d'accueil et le langage lu. Toujours est-il qu'il n'y aura pas pour autant contrefaçon du logiciel originel.

125. – Toutefois, l'interface qui permet de lire les données, dans l'affaire *Fiducial contre DPSI*³²⁹, et celle qui permet au logiciel d'accueil de comprendre les scriptes, dans l'affaire *SAS contre WPL*³³⁰, sont des programmes exprimés en code objet. L'étude du code objet est libre, mais pas sa reproduction. Le principal défaut de cette décision est qu'elle semble déduire de la non-décompilation du logiciel l'absence de contrefaçon par reproduction, alors que les deux actes sont indépendants. Il semblerait que l'ignorance de la forme du code source rende impossible la contrefaçon, ou présume son absence, alors que le code objet est tout autant protégé que le code source. Cette solution est à rapprocher d'une décision fort critiquable de la Cour de cassation rendue dans l'affaire *Mestiri contre Lecaye*³³¹, selon laquelle on ne saurait contrefaire une bible de roman policier qui n'a jamais été divulguée puisque le prétendu contrefacteur n'a jamais eu connaissance de son existence. Il semblerait même que la solution soit assez classique³³². Au contraire, la forme de l'interface n'est pas véritablement ignorée dans les affaires *Fiducial* et *SAS* puisqu'il y a étude des *inputs* et des *outputs*, donc du code objet. La société concurrente pratique un « surmoulage » sur ces comparaisons d'*inputs* et d'*outputs* pour le reproduire dans son propre logiciel. Ainsi, les codes sources divergent mais le code objet est pratiquement identique.

126. – Pourtant, l'interface n'a-t-elle pas une forme imposée par le langage de programmation qu'elle permet de lire de sorte que seule la contrefaçon par reproduction du langage puisse être reprochée ? C'est une étude au cas par cas qui permettra de déterminer si l'interface est originale par rapport à sa fonction qui est de lire le langage de programmation.

prévues aux II, III et IV du présent article est nulle et non avenue. ».

328 Cour du district de San Jose, 9 sept. 1998, *Sun contre Microsoft*. cf. *supra*, n° 104.

329 Cass. civ. 1^{re}, 20 oct. 2011, n° 10-14.069, F-P+B+I, *Sté Fiducial c/ Sté Développement professionnel spécialisé informatique*.

330 CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. c/ world programming Ltd.*

331 CA Paris, 4^e ch., 30 janv. 2008, *Mestiri c/ Lecaye*, Juris-Data n° 361193.

332 BERTAND (A.), « Chapitre 116.13. Principe » in *Droit d'auteur*, éd. Dalloz, 2013.

Il reste que cette faculté d'étude est strictement réservée à l'utilisateur légitime, ce qui aurait pu en réduire la portée si la cour n'avait pas écarté la faculté pour l'ayant droit d'interdire contractuellement l'étude à des fins de créations d'un logiciel concurrent. Il suffira au concurrent de se procurer une licence d'utilisation.

Conclusion

127. – Les langages de programmation sont les outils fondamentaux de conception des programmes, ils constituent le tissu de liaison entre les logiciels et avec le matériel. Véritables œuvres de l'esprit, parfois enregistrés comme marque ou comme invention, il n'en reste pas moins qu'ils ne se concurrencent que rarement, chaque génération devenant une espèce nouvelle aux propriétés communes mais divergentes de ses parents. Le pouvoir colossal de l'ayant droit sur l'univers informatique doit donc être davantage bridé que pour un simple logiciel.

En guise de conclusion, bien que la réflexion aurait pu avoir son siège au sein des limites au droit d'auteur, on peut se demander s'il n'y a pas un droit de modification et d'évolution des langages de programmation. Les programmes étant des œuvres utilitaires on ne saurait interdire à l'utilisateur légitime de le modifier afin qu'il soit conforme à sa destination, la maintenance corrective est expressément prévue par l'article 5.1 de la directive européenne 91/250 de 1991³³³. Mais qu'en est-il des maintenances évolutives ? À l'égard des logiciels on pourrait n'admettre que les maintenances adaptatives, c'est-à-dire les maintenances évolutives permettant d'adapter le programme aux nouvelles normes de sorte qu'il reste conforme à sa destination. Le logiciel étant une œuvre évolutive par nature, toute maintenance ne devrait-elle pas être autorisée ? Le débat est ouvert, il doit l'être bien plus encore à l'égard d'œuvres exclusivement tournées vers l'innovation comme les langages de programmation. Ceux-ci doivent s'adapter à l'évolution technologique des processeurs³³⁴ ainsi qu'à une demande de logiciels de plus en plus complexes et spécialisés. Interdire l'évolution des langages de programmation serait remettre en cause toute leur évolution darwinienne et serait une atteinte excessivement grave à l'innovation rompant ainsi l'équilibre fondamental entre le droit privé de l'auteur et l'intérêt public.

333 Art. 5.1 de la directive 91/250 : « *Sauf dispositions contractuelles spécifiques, ne sont pas soumis à l'autorisation du titulaire les actes prévus à l'article 4 a) et b) lorsque ces actes sont nécessaires pour permettre à l'acquéreur légitime d'utiliser le programme d'ordinateur d'une manière conforme à sa destination, y compris pour corriger des erreurs.* ».

334 Cf. *supra*, n° 11-17.

Sources et bibliographie

Sources

Textes officiels et normatifs :

BOT (Y.), Conclusions de l'Avocat général, CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. contre world programming Ltd.*

Code de la propriété intellectuelle, articles : L. 112-1 , L. 112-4 , L. 113-3, L. 113-4, L. 113-5, L. 122-6, L. 122-6-1 5, L. 122-7-1, L. 131-2, L. 131-3, L. 331-32, L. 335-2, L. 335-3, L. 611-10 2,

CSPLA avis n° 2005-1, 7 déc. 2005.

Directive 2001/29/CE, 22 mai 2001, sur l'harmonisation de certains aspects du droit d'auteur et des droits voisins dans la société de l'information.

Directive 91/250/CEE, 14 mai 1991, concernant la protection juridique des programmes d'ordinateur.

Directive 2009/24/CE 23 avril 2009, concernant la protection juridique des programmes d'ordinateur.

Directive 2011/83/UE, 25 oct. 2011, relative aux droits des consommateurs.

Décisions de justice :

Décision du CRT de l'OEB 3.5.1, 15 juil. 1986, T208/84, *Vicom*, JO de l'OEB 1987

CJCE, 6 avril 1995, décisions 241/91 et 242/91, *RTE et ITP Ltd c/ Commission*.

CJCE, 9 nov. 2004, C-203/02, *The British Horseracing Board Ltd e.a. Contre William Hill Organization Ltd.*

TPICE, 17 sept. 2007, T-201/04, *Microsoft contre Commission CE*, JO UE n° C279, 10 nov. 2007, p. 45.

CJUE, 16 juillet 2009, C-5/08, *Infopaq International*.

CJUE, 22 déc. 2010, C-393/09, *Bezpečnostní softwarová asociace*.

CJUE, 1^{er} mars 2012, C-604/10, *Football dataco*.

CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. contre world programming Ltd.*

Cass. AP, 7 mars 1986, *Pachot* ; *RTD Com.* 1986, p. 399, obs. FRANÇON (A.) ; *RID Admin.*, juill. 1986, p. 136, obs. LUCAS (A.).

Cass. Civ. 1^{re}, 12 avr. 1976, Bull. civ. I, n° 123.

Cass. civ. 1^{re}, 6 juin 1990, n° 88-11.811, *Textiles Maurice et autres contre SA Goutarel*.

Cass. civ. 1^{re}, 28 janv. 2003, n° 00-20.294, *Casaril contre Société Havas interactive*.

Cass. civ. 1, 17 juin 2003, n° 01-17.659.

Cass. civ. 1^{re}, 25 juin 2009, n° 07-20.387, *Cryo*.

Cass. civ. 1^{re}, 22 sept. 2011, n° 09-71.337 : JurisData n° 2011-019931 ; *RIDA*, 2011 n° 230, p. 149 et p. 249, obs. SIRINELLI (P.) ; *RTD Com.* 2011, p. 741, obs. POLLAUD-DULIAN (F.).

Cass. civ. 1^{re}, 20 oct. 2011, n° 10-14.069, F-P+B+I, *Sté Fiducial c/ Sté Développement professionnel spécialisé informatique*.

Cass. civ. 1^{er}, 17 oct. 2012, n° 11-21.641.

Cass. com., 12 juil. 2005, n° 04-12.388, *NMPP*.

Cass. com., 8 avr. 2008, n° 07-11.385.

Cass. crim., 18 oct. 2011, n° 11-81.404.

Cass. com., 14 fév. 2012, n° 10-27.873.

CA Paris, 14 janv. 1992, *Griolelet contre Vautrin*.

CA Paris, 4^e ch, 10 nov. 1994, *PIBD* 1995, 582, III, p. 27.

CA Paris 23 janv. 1995.

CA Paris, 4^e ch., 7 mars 2007, n° 06/02813, *Monopoly*.

CA Paris, 4^e ch., 30 janv. 2008, *Mestiri contre Lecaye*, Juris-Data n° 361193.

CA Paris, 4^e ch., 24 sept. 2008.

CA Paris, 4^e ch., 1^{er} oct. 2008, *Studio 15 Beauté contre DF France, Éole et Dessange International*.

CA Paris p. 6, 1^{re} ch., 17 oct. 2012 ; com. par : BRUGIÈRE (J.-M.), in « Droit d'auteur et droits voisins » (chronique), *Propriétés intellectuelles* n° 46, janvier 2013, p. 43-44.

Décision du Conseil de la concurrence, 9 nov. 2004, déc. n° 04-D-54, *Apple iTunes contre VirginMega*.

TGI Paris, ord. réf., 5 sept. 2011, *SAS Universal music France contre SAS Blogmusik*.

Cour du district de San Jose, 9 sept. 1998, *Sun contre Microsoft*.

Cour du district Nord de Californie, 23 mai 2012, C10-03561 WHA, *Oracle contre Google*.

High court of Justice, Chancery Division, 25 janv. 2013, *SAS Institute versus WPL*.

Bibliographie

Monographies non juridiques :

- BELLE (Martine) et LORET (Patrice), *L'outil informatique, pour les sciences et les sciences économiques . Des langages aux applications*, Masson, Paris, 1987
- CHENIQUE (François), *Software . Langages et systèmes d'exploitation*, Dunod, Paris, 1971
- CHOMSKY (Noam), *Le langage et la pensée*, Syntactic Structures, Paris, 1957
- DRÉAN (Gérard), *L'industrie informatique . Structure, économie, perspective*, Masson, Paris, 1996.
- GRELON (Bernard), *Les entreprises de services*, Economica, 1978.
- GRENIER (Jean-Guy), *Dictionnaire d'informatique et d'internet . Anglais-français*, La maison du dictionnaire, 1996.
- HIGMAN (Bryan), traducteur MARGULLICI (M.-T.), *Étude comparative des langages de programmation*, Dunod, 1973.
- HORN (François), *L'économie du logiciel*, thèse soutenue le 18 décembre 2000, Lille, accessible sur : <http://www.univ-lille1.fr/bustl-grisemine/pdf/extheses/50374-2000-23-24.pdf>.
- KLOETZER (Henri), *Introduction à l'économie numérique*, Lavoisier . Management et informatique, Cachan, 2012.
- POYEN (Jacques) et POYEN (Jeanne), *Le langage électronique*, PUF . Que sais-je ?, 3^e éd., 1967.
- POYEN (Jacques) et POYEN (Jeanne), « Langage et automatisation » in *Points de vue sur le langage*, Éd. Klincksieck, Paris, 1969
- REY (Alain), REY-DEBOVE (Josette), ROBERT (Paul), *Le nouveau petit Robert de la langue française*, Le Robert, 2008.
- SHUMPETER (Joseph), *Capitalisme, socialisme et démocratie*, Petite Bibliothèque Payot, 1974.

Monographies juridiques :

- BASALAMAH (Salah), *Le droit de traduire . Une politique culturelle pour la mondialisation*, Les Presses de l'Université d'Ottawa, 2009
- BERTAND (André), « Chapitre 116.13. Principe » in *Droit d'auteur*, éd. Dalloz, 2013.
- BERTAND (André), « Chapitre 202 Logiciels » in *Droit d'auteur*, éd. Dalloz, 2013.
- BIGOT (J.-P.), « Détecter et matérialiser la contrefaçon d'un logiciel : la méthode américaine AFC Test, et l'outil logiciel de ESALAB », *CCE* n° 12, déc. 2006, étude 38.
- BREESE (Pierre) et KERMADEC (Yann), *La propriété intellectuelle au service de l'innovation*, Nathan . INPI . Repères pratiques, Paris, juil. 2009.

- BRUGUIÈRE (Jean-Michel) et VIVANT (Michel), *Droit d'auteur et droits voisins*, Précis Dalloz, 2012
- CARON (Christophe), *Droit d'auteur et droits voisins*, Litec . LexisNexis, 2^e éd., 2009
- CHEN (Szu-Ting), *Droit d'auteur et protection technique des œuvres – étude de droit comparé européen et asiatique*, thèse soutenue le 13 déc. 2010, Nantes.
- DERCLAYE (Estelle) et STROWEL (Alain), *Droit d'auteur et numérique : logiciels, bases de données, multimédia*, Bruylant, 2001, Bruxelles.
- DESBOIS (Henri), *Le droit d'auteur en France*, Dalloz, 3^e éd., 1978, Paris.
- GAUTIER (Pierre-Yves), *Propriété littéraire et artistique*, Puf, 8^e éd., 2012.
- HOLLANDE (Alain) et LINANT DE BELLEFONDS (Xavier), *Pratique du droit de l'informatique*, Delmas, 5^e éd., 2002, p. 186, n° 1103.
- LAMZOUHRI (Adam), *L'incidence de la protection du droit d'auteur sur le logiciel*, mémoire, 21 mai 2012, Paris-Ouest, accessible sur : <http://www.e-juristes.org/wp-content/uploads/2012/05/Lincidence-de-la-protection-du-droit-dauteur-sur-le-logiciel.pdf>.
- LAPORTE-LEGEAIS (Marie-Eugenie), « Traductions juridiques et droits intellectuels » in *Traduction du droit et droit de la traduction*, sous la dir. de CORNU (Marie) et MOREAU (Michel), Dalloz, 2011.
- LATREILLE (Antoine), *Les mécanismes de réservation et les créations multimédias*, thèse soutenue le 20 déc. 1995, Paris-Sud.
- LINANT DE BELLEFONDS (Xavier), « Les outils logistiques du net et la propriété intellectuelle », in *Internet saisi par le droit*, édition des Parques 1997, p. 147
- LINANT DE BELLEFONDS (Xavier), *Droits d'auteur et droits voisins*, Dalloz . Cours, 2^e éd., 2004.
- LUCAS (André), LUCAS (Henri-Jacques) et LUCAS-SHLOETER (Agnès), *Traité de la propriété littéraire et artistique*, 4^e éd., 2012, LexisNexis.
- LESCURE (Pierre), *Mission culture – Acte 2 . Contribution aux politiques culturelles à l'ère numérique, rapport au ministère de la culture* [en ligne], mai 2013, accessible sur : http://www.culturecommunication.gouv.fr/var/culture/storage/culture_mag/rapport_lescore/index.htm.
- MACREZ (Franck), *Créations informatiques : bouleversement des droits de propriété intellectuelle ? Essai sur la cohérence des droits*, Litec . Collection du CEIPI, 2011.
- POLLAUD-DULLIAN (Frédéric), *Droit d'auteur*, Economica . Corpus, 2005.
- RELMY (Jean-Pierre), *Le droit de la traduction . Contribution à l'étude du droit du langage*, thèse soutenue en 2007, Paris-Sud.

SANTOPAULO (Marc-Antoine), *Protection juridique des langages de programmation*, Mémoire DEA 2002, Montpellier.

SCOT D. (Michael), *Computer Law*, Wiley law publications, Etats-Unis, 1984.

STÉRIN (Anne-Laure), *Guide pratique du droit d'auteur*, Maxima . Concrètement que dois-je faire, Paris 2011, 2e éd.

VIVANT (Michel), « 160 - À propos des langages de programmation », in *Lamy Droit du Numérique*, Lamy, 2012.

Articles, notes, publications juridiques :

BERNAULT (Carine), « Étendue de la protection des logiciels par le droit d'auteur », *L'essentiel . Droit de la propriété intellectuelle*, 15 juil. 2012.

BENABOU (Valérie-Laure) note sous CJUE, 22 déc. 2010, C-393/09, *Bezpenostní softwarová asociace*, *Propriétés Intellectuelles*, 2011, 39, p. 205.

BOUCHE (Nicolas), com. sur CJUE 2 mai 2012, *Propriété industrielle* n° 7, juil. 2012, comm. 61.

BRUGUIÈRE (Jean-Michel), « Protection du titre d'un jeu (non) » in « Droit d'auteur et droits voisins » (chronique), *Propriétés intellectuelles* n° 46, janv. 2013.

CHAMPIGNEULLE-MIHAILOV (Jeanne), « Les conflits relatifs à l'utilisation des normes par les professionnels », *Petites affiches*, 11 février 1998, n° 18, p. 21.

DESBOIS (Henri), note sous Trib. Comm. Seine, 26 juin 1951, *Les hauts de Hurlevent*, *RTD Com.* 1952, p. 763, §10.

GAUDRAT (Philippe), « L'invention informatique : un débat difficile et contourné », *RTD Com.*, 2005, p. 512.

HUILIER (Jacques), « Le parasitisme parasite-t-il la propriété intellectuelle ? », *Gazette du Palais*, 7 nov. 2001, p. 6.

LATREILLE (Antoine), « La création multimédia comme œuvre audiovisuelle », *La Semaine Juridique*, éd. générale, n° 31, 29 juil. 1998, I 156.

LE STANC (Christian), « Droit de l'informatique », *La semaine juridique entreprise et affaires* n° 21, 23 mai 1996, 559.

MACREZ (Franck), « Le droit d'auteur, le programme d'ordinateur et la cour de justice », *RIDA*, 22 octobre 2012.

MACREZ (Franck), « Le logiciel chez les Bretons (le retour) », *RLDI*, 1^{er} avril 2013.

MARINO (Laure), « Chronique de jurisprudence de Propriété littéraire et artistique », *Gazette du Palais*, 2 août 2012, n° 215.

MARTIN (Jean-Pienne), « La protection des logiciels informatiques : droit d'auteur ou brevet

d'invention ? », *La Semaine Juridique Entreprise et Affaires* n° 16, 25 avr. 1990, 15752

POLLAUD-DULLIAN (Frédéric), note sous CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. contre world programming Ltd*, *RTD Com.*, 2012, p. 536.

ROUSSEL (Patrick), « La maîtrise d'un langage de programmation s'acquiert par la pratique », *CCE* n° 4, avril 2005, étude 15, n° 1, §1.

SARDAIN (Frédéric), « Hors-piste n° 1 » in « Propriété littéraire et artistique », *La semaine juridique entreprise et affaires* n° 28, 12 juil. 2012, 1461.

SIRINELLI (Pierre), note sous CJUE, 2 mai 2012, C-406/10, *SAS Institute Inc. contre World Programming Ltd*, *RIDA*, juil. 2012, p. 209 et s.

SIRINELLI (Pierre), « Propriété littéraire et artistique », *Recueil Dalloz* n° 42, 2012, p. 2836.

TREPPOZ (Édouard), « La qualification logicielle d'un jeu-vidéo : un modèle pour les œuvres multimédia », *Les petites Affiches*, 230, 18 nov. 1999.

ZÉRAOUI-SARAH (Farha) « Les traductions : le régime de protection par le droit d'auteur . Étude comparative droit algérien-droit français », *Propr. intel.* n° 46, janv. 2013, p. 30-42.

Articles et ressources en ligne libres d'accès :

ANONYME, logiciel de conversion des caractères en binaire, consulté le 15 mars 2013 : <http://www.convertbinary.com/>

ANONYME, description du brevet « Ordinateur quantique », consulté le 10 mars 2013 : <http://www.patfr.com/200109/WO0171661.html>

BEAUCHAMP-TREMBLAY (Xavier), « Droit d'auteur dans les logiciels 101 – Oracle c. Google », *ActifsCreatifs* [en ligne], 12 déc. 2012, consulté le 28 mai 2013 : <http://www.actifscreatifs.com/droit-auteur-logiciel-oracle-google/>

FAVRE (Denis) « L'invention de la mécanographie », *interstices* [en ligne], http://interstices.info/jcms/n_51928/linvention-de-la-mecanographie, consulté le 29 mai 2013.

FROCHOT (Didier), « Du nouveau sur les normes et sur l'AFNOR », *Les infostrateges* [en ligne], 26 juin 2009, consulté le 5 mai 2013 : <http://www.les-infostrateges.com/actu/0906742/du-nouveau-sur-les-normes-et-sur-l-afnor>

HUNAUULT (Gilles), *[petite] histoire des langages de programmation* [en ligne], <http://www.info.univ-angers.fr/~gh/hilapr/detour.htm>, consulté le 28 mai 2013.

KINNERLEY (Bill), *The language list*, [en ligne], <http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm>, consulté le 22 mai 2013.

LÉVÉNEZ (Éric), *Computer languages history*, [en ligne], <http://www.levenez.com/lang/>, consulté le 22 mai 2013.

Table des matières

Remerciements.....	3
Introduction.....	5
Chapitre 1 :La valeur considérable des langages de programmation.....	8
I)L'opportunité économique et technique favorable à la protection des langages.....	8
A)Le langage : une étape technique améliorant la conception des logiciels.....	8
1.Le langage : une solution technique ayant facilité la conception moderne des logiciels....	8
2.Protection d'investissements et valeurs économiques conséquents.....	13
B)Le langage : une étape technique indispensable à la conception moderne.....	17
1.Les langages incontournables.....	17
2.Les arguments libéraux opposés à la réservation des langages et des formats.....	19
II)La protection principale des langages de programmation par le droit d'auteur.....	22
A)La vision de surface des langages de programmation : une règle abstraite agrémentée d'éléments protégés.....	23
1.Le langage : une œuvre agrémentée d'éléments originaux.....	23
2.Le langage : plus qu'une simple théorie de la programmation.....	27
B)La véritable nature du langage : une programmation fondamentale à l'usage réglementé...31	
1.Le langage : un programme préparatoire dans la conception de logiciels.....	31
2.La syntaxe : une règle conditionnant l'utilisation.....	34
Chapitre 2 :Le régime juridique complexe des langages de programmation.....	39
I)D'un régime distributif vers un régime unitaire préférable.....	39
A)Une qualification éclatée contestable.....	39
1.Principe de l'éclatement et distribution mal aisée.....	39
2.Qualification recentrée sur l'œuvre composite ou le cœur de l'œuvre.....	42
B)Des mécanismes intéressants pour l'unité à atteindre.....	44
1.L'intérêt mitigé de l'unité face à l'inacceptable éclatement.....	44
2.Des montages juridiques plutôt envisageables.....	47
II)Une exploitation multiforme et raisonnée des langages de programmation.....	50
A)Les modes et modèles d'exploitation : le libre choix de l'auteur.....	50
1.Les trois modes d'exploitation incontournables.....	50
2.Les modèles économiques : du rejet du privatif aux pièges du libre.....	54
B)Le langage : un droit limité et une entrave aux exceptions.....	57
1.Droit de la traduction des programmes.....	58
2.Droit d'étudier le logiciel à des fins d'interopérabilité et de création.....	61
Conclusion.....	66
Sources et bibliographie.....	67
Table des matières.....	73