

UN RÉGIME
DE DROIT D'AUTEUR :
la propriété intellectuelle
du logiciel

Jean-Benoît ZIMMERMANN

A l'origine de l'industrie informatique, le logiciel (software) était considéré comme indissociable du matériel (hardware), avec lequel il était fourni et naturellement facturé de manière jointe. Développé par le constructeur ou en interne par l'utilisateur, pour ses propres besoins d'applications, il ne donnait pas naissance à un secteur marchand spécifique, si ce n'est toutefois comme part de l'activité des « sociétés de services et de conseil en informatique » qui développaient, à la demande et de manière spécifique, des logiciels d'application sur mesure pour certains gros utilisateurs.

Au fil de l'évolution de l'industrie informatique et des concepts qui ont marqué son histoire, le logiciel s'est peu à peu émancipé et a gagné sa propre autonomie, sur les plans économique (production, marché) et technique (portabilité). Il devient non seulement complémentaire, mais aussi fondamentalement substitutif des architectures matérielles. Celles-ci évoluent vers des concepts à plus grand

degré d'universalité, tandis que le logiciel revêt une fonction d'ordonnement des ressources matérielles en vue d'un champ d'utilisation spécifié. Le logiciel, dans cette évolution, d'une logique câblée à une logique programmée, devient dès lors le facteur clef de la flexibilité des architectures de traitement de l'information (pris cette fois dans un sens plus large que celui des seuls ordinateurs). Il prend de ce fait une importance grandissante en part relative sur les marchés (1), mais aussi, d'un point de vue technique, dans l'architecture des systèmes d'information.

Le fait est que, concernant le statut marchand du logiciel, on a encore souvent à faire à une certaine confusion. Il est ainsi significatif de constater une fréquente assimilation des logiciels et des services (informatiques) comme relevant d'une seule et même activité. La nomenclature des activités en France par exemple (code APE) ne reconnaît pas le logiciel comme une activité industrielle (contrairement au matériel bien sûr et malgré son aspect complémentaire et substitutif). Le fait est que le développement d'un logiciel en réponse à une demande formulée comme un « problème » de traitement de l'information peut être considéré comme un service visant à permettre à l'utilisateur de tirer parti d'un système informatique dans un but donné. Il conviendrait tout au moins de reconnaître la distinction entre un logiciel écrit sur mesure pour un client unique et un *progiciel* (2) vendu à de multiples exemplaires en raison de sa « portabilité ». Encore que la conception à façon inclue, de manière croissante, une certaine forme de standardisation relative à l'application de méthodes d'analyse et de développement et à l'utilisation ou à la réutilisation de modules standards (routines ou algorithmes, outils de génie logiciels...)

(1) On a, à ce propos, parlé d'une invention de la « règle des 80-20 », qui caractérisait les parts relatives du matériel et du logiciel à l'ère de la grande informatique. Le côté radical d'une telle affirmation peut certes paraître abusif et surtout peu rigoureux ; il n'en demeure pas moins que l'évolution comparée des dépenses fait apparaître un différentiel croissant en faveur du logiciel. En outre, les constructeurs d'ordinateurs, voyant leurs marges bénéficiaires peu à peu grignotées par une guerre de prix sans merci, cherchent à réaliser une part croissante de leurs activités dans le logiciel et les services, encore modérément affectés par la concurrence par les prix.

(2) On définit un progiciel comme un produit logiciel commercialisé de façon autonome et en standard, ce qui suppose un certain degré de portabilité.

Avec l'accès au statut de bien marchand du logiciel se posait, dès lors qu'il faisait l'objet d'une facturation séparée (3), la question de sa protection en tant que produit d'une activité d'invention et de création. Or le problème de la propriété intellectuelle constitue, en ce qui concerne le logiciel, un cas d'espèce en raison de sa nature particulière de bien marchand d'un genre nouveau. Expression intellectuelle, un logiciel n'est cependant pas assimilable à sa seule expression. Deux programmes similaires peuvent n'avoir qu'un faible niveau d'expression commune. La « performance » d'un programme, sa « qualité », ne transparaissent pas explicitement dans son expression. En tant qu'expression, il n'est qu'une expression arbitraire d'idées qui ne sont pas explicitement livrées et ne peuvent être accessibles qu'au terme d'un travail de décryptage et d'analyse. Au-delà de cette expression, il vaut par ses fondements qui lui confèrent une nature de processus : une logique et des principes de résolution ou d'action, une architecture, une organisation séquentielle, un ensemble de routines et d'algorithmes.

Nous rendons compte tout d'abord du débat relatif au choix ou à la construction d'un cadre juridique approprié aux questions de propriété intellectuelle dans le domaine du logiciel. Puis nous examinons la question du recours au système des brevets, comme alternative ou complément du droit d'auteur. Nous étudierons ensuite l'étendue et le niveau d'un « bon système de protection », en tenant compte des difficultés de la jurisprudence à apprécier la similarité de deux logiciels. Cette question nous conduira à prendre en compte les effets d'externalités de réseau, qui jouent un rôle grandissant dans cette industrie et dont le plein développement dépend de la liberté d'accès aux interfaces, afin de permettre le développement de produits complémentaires. Nous voyons que sous ces conditions, une approche de la standardisation du type de celle prônée dans les

« systèmes ouverts » est susceptible de fournir une clef de dépassement du « dilemme schumpétérien » entre retour sur investissement pour l'innovateur et dynamique industrielle.

La question d'un cadre juridique approprié de protection

Pour Lucas (4), « le programme, qui est au cœur du logiciel, se caractérise d'abord par son contenu comme un procédé permettant de tirer parti des ressources de la machine en vue d'un résultat déterminé. À ce titre, sa protection pose un problème de brevetabilité. Mais en même temps, il se présente en lui-même, apparemment au moins, comme une œuvre de l'esprit susceptible de donner prise au droit d'auteur [] De là vient qu'(il) puisse *a priori* prétendre à la fois au bénéfice du brevet d'invention et à celui du droit d'auteur. De là vient aussi qu'il ne puisse trouver commodément sa place ni dans l'une ni dans l'autre de ces deux branches du droit de la propriété industrielle, ce qui fait toute la différence en la matière. »

Fallait-il concevoir un cadre juridique spécifique pour un tel produit qui est à la fois technologie et expression ? Ce débat, qui a eu lieu dès les années soixante-dix dans les différents pays concernés, devait déboucher d'une façon générale, avec des variantes propres au droit et aux jurisprudences nationales, sur une adaptation des outils existants, avec une prédominance accordée au droit d'auteur. Certes la position américaine a joué d'un poids important dans cette orientation, mais la question posée a été également celle de la difficulté d'harmonisation des législations nationales et de la lourdeur d'une négociation de nouvelles conventions internationales. En outre le risque eût été grand de voir un cadre juridique, créé de toute pièce, rendu caduque avant même d'être devenu opératoire, par le biais d'un changement technique accéléré.

(3) Le principe d'une facturation séparée s'était imposé à partir de la fin des années soixante comme une conséquence des poursuites engagées par l'administration américaine, puis par les communautés européennes, contre IBM au nom de la législation antitrust et de l'abus de position dominante.

(4) LUCAS, 1987.

Aux États-Unis, « la CONTU (*National Commission on New Technological Uses of Copyrighted Works*) tient comme établi que, dans le proche futur, la plupart des infractions au copyright relèveront encore de la simple copie, mais la commission reconnaît que les progrès techniques soulèveront dans l'avenir des difficultés pour cerner l'étendue exacte de la couverture du copyright » (5) Mandatée par le Congrès, la CONTU a recommandé dans son rapport de 1978 la prise en compte du logiciel dans le champ juridique du copyright Ses recommandations se sont traduites par une série d'amendements (*Computer Software Copyright Act*) apportés en 1980 au *Copyright Act de 1976* Mais la commission était dès lors consciente des difficultés à venir dans l'application de la loi

En Europe, la Communauté Européenne est parvenue à l'élaboration d'un compromis entre les intérêts divergents des concepteurs, des constructeurs et des utilisateurs, sous la forme d'une Directive, votée par le Parlement le 17 avril 1991, adoptée à l'unanimité par le Conseil des Ministres les 12 et 14 mai et qui devait entrer en vigueur pour tous les pays membres le 1^{er} janvier 1993 Avec un objectif premier affiché de lutter contre le piratage, la Directive européenne autorise la copie de sauvegarde, la correction d'erreurs dans le cadre de l'utilisation prévue et, sous certaines conditions, la décompilation dans une perspective d'interopérabilité des programmes Nulle mention n'est faite de la propriété des logiciels commandés par une entreprise-cliente, ce qui comble les SSII (6) mais inquiète fortement les grandes entreprises utilisatrices

Ainsi en France, la loi sur la propriété des œuvres littéraires ou artistiques date du 11 mars 1957 La jurisprudence en avait longtemps exclu les programmes d'ordinateurs jusqu'au jugement de la 4^e Chambre de la Cour d'Appel de Paris le 2 novembre

1982, qui avait accordé un « droit d'auteur » à un informaticien, relativement à un programme qu'il avait conçu La jurisprudence, à travers des arrêts contradictoires, ne parvenant pas à fixer une orientation, le gouvernement, pressé par les milieux professionnels, élabore le contenu de la loi du 3 juillet 1985, ajoutant les logiciels à la liste de l'article 3 de la loi de 11/03/1957, qu'il étendait ainsi à la « protection » des logiciels « originaux » Mais ce texte ne levait pas une double incertitude, tenant d'une part à la notion d'originalité et d'autre part au champ de la protection La directive s'est traduite en France par l'adoption de la loi n° 92-597 du 1^{er} juillet 1992, modifiée par la loi n° 94-361 du 10 mai 1994 (7), qui permettait dès lors la définition d'un « code de la propriété intellectuelle » visant à « répondre à la nécessité de regrouper en un corpus unique l'ensemble des dispositions relatives aux droits d'auteurs et droits voisins d'une part et ceux relatifs à la propriété industrielle d'autre part » (8)

Au Japon, en revanche, la perception d'une valeur intrinsèque attachée à des produits immatériels, comme le logiciel, est relativement récente Elle s'est formée à travers un nombre important de poursuites dont des entreprises japonaises ont fait l'objet pour violation de droits de propriétés relatifs à des logiciels de micro-informatique, à la fois professionnels et de jeux Elle est surtout datée de la poursuite entamée par IBM à l'encontre de Hitachi et Fujitsu relativement à l'utilisation illécite de son système d'exploitation dans des ordinateurs compatibles

Dans deux études séparées, menées au cours des années 1983-1984, sur la question de la protection du logiciel, le MITI arrivait à la conclusion d'une nécessaire élaboration d'un cadre juridique spécifique, tandis que l'ACA (*Agency for Cultural Affairs*) préconisait une révision de la législation du copyright À l'issue d'un débat marqué par la sensibilité au contexte international ce fut

(5) OTA, 1992, p. 6

(6) Sociétés de service et d'ingénierie informatique

(7) « La protection des logiciels, code de la propriété intellectuelle (Loi n° 92-597 du 1^{er} juillet 1992) », Journal Officiel de la République française, 21 mai 1994

(8) Projet de loi, cité par BERTRAND, 1994, p. 23

cette dernière solution qui fut retenue (9) Reste à noter cependant que la nouvelle loi japonaise sur le copyright exclut de la liste des œuvres protégées les langages de programmation ainsi que les règles et algorithmes et qu'elle maintient le droit à la modification des programmes, pour en améliorer l'efficacité ou les adapter à un système non compatible

Le recours alternatif au droit des brevets

Si la législation du copyright et du droit d'auteur a été partout retenue comme le cadre privilégié de la protection du logiciel, il n'en demeure pas moins que d'autres cadres juridiques alternatifs et complémentaires demeurent accessibles et notamment le recours aux brevets qui soulève un problème de fond quant à la dynamique propre de l'industrie logicielle

En France, les dispositions relatives au droit d'auteur peuvent se cumuler à la protection des marques, à celle des dessins et modèles, pour les aspects graphiques d'un logiciel, et surtout à la protection des brevets, dans la mesure où le logiciel concerné présente un aspect de processus industriel En réalité, l'applicabilité de la loi sur les brevets d'invention du 2 janvier 1968 (qui renouvelait l'ancienne loi datant de 1844), Article 7, excluait « les programmes ou séries d'instructions pour le déroulement des opérations d'une machine calculatrice » Plus que pour des raisons juridiques, on craignait alors de voir IBM s'engouffrer dans une brèche législative qui lui aurait permis d'asseoir sur le long terme une position dominante dans l'hexagone qui, contrairement aux États-Unis, ne disposait pas d'arsenal juridique antitrust La loi du 13 juillet 1978, qui visait à harmoniser la législation française avec le texte de la Convention de Munich (1973), confirmait cette restriction

Sur cette question la jurisprudence française s'est avérée assez stricte avant

d'admettre cependant comme brevetables (Affaire Schlumberger, Cour d'Appel de Paris 15/06/81) certains logiciels, à condition toutefois qu'ils soient présentés non pas en eux-mêmes mais comme parties intégrantes et inséparables de processus industriels, entendus dans le sens le plus strict du terme « Un programme informatique est indirectement brevetable aux conditions de fond suivantes il doit constituer une étape d'élaboration d'un procédé industriel brevetable, il doit satisfaire lui-même aux trois critères de brevetabilité la nouveauté, le résultat industriel et l'activité inventive (10) » Il n'en demeure pas moins que, contrairement à la jurisprudence américaine, la protection reste limitée au cadre du procédé de référence (brevet principal), « il est probable que l'utilisation de ce programme, dans une toute autre fin que celle pour laquelle le brevet a été déposé, reste libre » (11)

Toutefois les directives du 6 mars 1985 de l'Office Européen des Brevets admettent la brevetabilité d'un logiciel qui apporte « une contribution de caractère technique à l'état de la technique », à l'instar d'un programme qui, chargé dans un ordinateur, en modifie le comportement En outre, le programme revendiqué doit « faire l'objet d'une description fonctionnelle suffisante afin de pouvoir être exécuté par un homme de l'art »

Aux États-Unis la protection du logiciel, à la différence des autres œuvres intellectuelles, s'appuie sur une triple dimension législative copyright, brevets et secret de fabrication La loi sur le copyright protège l'expression d'une idée plutôt que l'idée sous-jacente elle-même Le brevet quant à lui protège les droits de propriété sur la manière de faire quelque chose Le secret de fabrication enfin, comme le brevet, protège l'idée sous-jacente d'une invention plutôt qu'une expression particulière, mais par contre présente pour son détenteur l'avantage de ne pas nécessiter une quelconque divulgation d'information Il ne confère pas, en revanche, de monopole d'exploitation La position de départ du *Patent and Trademark*

(9) YAMAMOTO, 1989

(10) VAN DORSSELAERE, 1987

(11) Idem

Office (PTO) était hostile à la délivrance de brevets pour des programmes d'ordinateurs, en raison de leur nature assimilable à un enchaînement d'étapes abstraites et/ou d'algorithmes mathématiques. Une succession d'actions, notamment devant la Cour Suprême et devant la *Court of Customs and Patent Appeals* (CCPA) a conduit à une révision de la doctrine de départ et une certaine pratique de délivrance de brevets pour des « composants » logiciels.

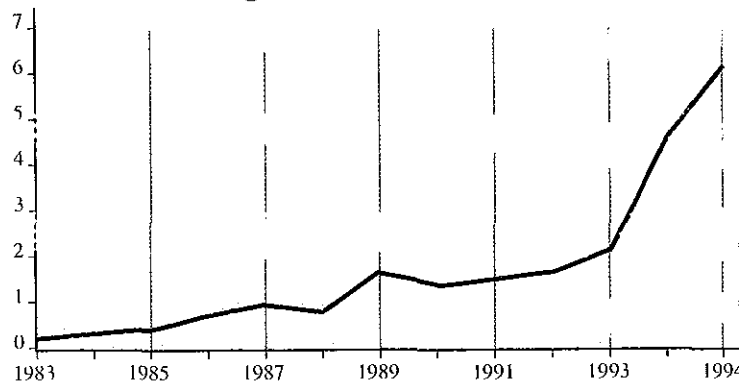
Au cours des années 80 on a pu assister à une croissance spectaculaire du nombre d'entrepreneurs aux États-Unis qui obtiennent de protéger leurs logiciels par des brevets, espérant ainsi améliorer la protection de leurs droits, comparativement à une législation fondée sur le droit d'auteur, jugée trop étroite.

S'il s'agit là pour les entrepreneurs d'un réflexe de défense pour faire front aux pirates et aux « cloneurs » en tous genres, cette pratique et son institutionnalisation risquent de porter un coup d'arrêt à la dynamique de l'innovation, enfermant chaque

producteur derrière des murailles élevées et rendant de ce fait impossible toute pratique combinatoire. Le fait est qu'une multitude de brevets déposés correspondent à des procédures, voire des algorithmes utilisables au sein d'un programme d'ordinateur. Beaucoup d'entre eux, loin des critères de nouveauté et d'originalité, apparaissent de nature conventionnelle ou évidente aux yeux des développeurs, qui craignent en conséquence de subir une pluie de procès individualisés, relatifs à telle ou telle procédure communément employée dans un logiciel.

« Le problème se combine au fait que les suites logicielles modernes sont composées de milliers de processus brevetables, ajoutant chacun au risque d'empiéter sur un brevet en cours de protection. Dans la mesure où les applications logicielles sont interdépendantes et doivent être soigneusement articulées les unes aux autres, les développeurs informatiques risquent d'éprouver des difficultés à supprimer un processus qui ferait partie intégrante d'un programme original (12) »

Nombre de demande de brevets dans le domaine du logiciel USA 1983-1995 (milliers)



Source: *Business Week*, « *US Patent Corporation ?* », October 30, 1995

(12) KAHIN, 1990

D'un côté et au-delà des abus effectifs (13), on comprend que même si la demande de protection avait concerné, au départ, le champ des produits, elle dérive assez naturellement, par le fait de certaines pratiques de revendication, sur le champ de la production. Et ceci d'autant plus que la complexification des logiciels implique et a été rendue possible par le recours à des méthodes modernes de programmation structurée. Celles-ci permettent de construire des programmes sur la base d'une combinaison de modules élémentaires incorporés au sein d'une architecture d'ensemble. Il en dérive une certaine codification et une amorcée d'industrialisation dans la production des logiciels, fondée sur la systématisation du recours à des « composants logiciels » portables et réutilisables.

D'un autre côté, l'émergence de tels composants correspond également à un recours accru aux fondements mathématiques de la programmation. Et l'on est renvoyé de manière de plus en plus systématique à la question de la distinction entre bien public et bien privé en ce qui concerne ces modules et algorithmes. La tendance antérieure était de considérer que de telles élaborations intellectuelles ont, de par leur nature même, un caractère de non-rivalité qui en fonde l'accessibilité généralisée. Les législations du droit d'auteur ont d'ailleurs clairement exclu du champ de la protection les principes et algorithmes. L'octroi de brevets crée en revanche une barrière à l'utilisation et interdit donc l'idée de non-exclusion de tels biens, alors même que les pratiques en cours dans l'industrie s'appuyaient sur la cumulativité et l'interactivité. En outre les critères de nouveauté et d'originalité, attachés au système des brevets, paraissent ici difficiles à fon-

der en raison de l'absence d'un inventaire de l'état de l'art et mais aussi de pratiques antérieures de dépôt de brevets.

Rares sont ceux des développeurs qui s'étaient jusque-là souciés de s'informer de l'existence d'éventuels brevets sur de telles entités abstraites. « Traditionnellement, les « procédés » comprenaient les méthodes de transformation physique des matériaux et des objets, mais ne prenaient pas en compte les procédures de gestion ni les raisonnements mentaux. De ce fait, les programmes informatiques se situent quelque part entre le territoire traditionnel du copyright et le brevet (14) ». Une telle dérive risque de freiner la dynamique de l'innovation dans un domaine particulièrement sensible aux externalités de réseau et aux innovations incrémentales. Le risque est grand d'entraîner un net ralentissement de la compétitivité américaine dans le domaine du logiciel. En outre les premières victimes d'une vague de procès dans ce sens seront les petites entreprises qui n'auront pas les moyens de contester en justice et devront accepter de payer des droits relativement à ces revendications (15). À long terme c'est la structure même de l'industrie qui risque d'en être bouleversée et le prix des logiciels qui risquent d'en souffrir.

Sensible à de tels arguments, la justice américaine a été très prudente dans l'application du droit des brevets au domaine du logiciel, la Cour Suprême s'en tenant au principe de lier les algorithmes informatiques à des « lois de la nature » et des principes scientifiques et allant jusqu'à invalider certains brevets. Néanmoins, pour la première fois en 1994 Microsoft s'est vu condamné pour violation de brevets de compression de données détenus

(13) À la fin des années quatre-vingt, Refac, entreprise engagée dans une logique de valorisation systématique de droits de propriété acquis, a intenté une série de poursuites à l'encontre de six grands éditeurs de logiciels « tableurs » parmi lesquels Lotus. L'objet du litige reposait sur un brevet concernant une technique dite « *natural order recalc* » qui consiste à recalculer les conséquences sur l'ensemble du tableau d'un changement de contenu de l'une des cellules de la feuille de calcul. Ce principe technique est utilisé par la très grande majorité des programmes de ce type.

(14) KAHIN, 1990.

(15) IBM à elle seule a par exemple déposé 1 087 brevets aux États-Unis en 1993, dont 20 % concernaient des logiciels. Aujourd'hui, elle entend clairement réaliser la rente représentée par son portefeuille et réclame des royalties auprès de nombreuses firmes de logiciel parmi lesquelles Oracle, Computer Associates, Adobe, Autodesk, Intuit, Informix. (« Logiciel IBM part à la pêche de ses royalties », 01 Informatique, N° 1444, 14 mars 1997). En outre, les procès en chaîne qui se déroulent sans cesse devant la justice américaine mettent en œuvre des armées d'avocats.

par Stac, avec laquelle elle avait échoué à s'entendre sur les conditions d'un accord de licence. Microsoft a été condamnée à verser 120 millions de dollars de dommages et intérêts à Stac et contrainte de retirer provisoirement le produit en cause du marché (16)

Reste que le recours ou non au système des brevets traduit les deux faces d'un paradoxe rendu d'autant plus sensible que l'on se situe dans un domaine technologique caractérisé à la fois par une forte dynamique de l'innovation et l'importance des aspects cumulatifs du changement technique (17). Ainsi les partisans des brevets insistent sur les dommages subis par le biais d'une imitation de leurs propres effets de développement et de créativité. C'est l'incitation à l'innovation qu'il s'agit de défendre de manière plus efficace. Les défenseurs du droit d'auteur quant à eux mettent en avant l'importance des progrès qu'un concept peut connaître par l'intervention successive de développeurs nouveaux, à l'exemple du traitement de texte ou de la feuille de calcul, dont un brevet accordé aux inventeurs aurait bloqué ou tout au moins fortement freiné l'évolution technique et des prix. Ils opposent une logique d'innovation à une logique de rentes, dans laquelle les entreprises seraient conduites à davantage se battre devant les tribunaux que sur les marchés (18).

L'étendue et le niveau de la protection

Si la nature de bien marchand qu'a acquis le logiciel implique de disposer des outils nécessaires à sa protection, c'est qu'il s'agit pour les développeurs de faire face à deux périls : le piratage et l'imitation. Mais au-delà, ce sont souvent des positions dominantes qui sont recherchées, dans une optique de concurrence monopolistique.

Le phénomène du piratage, tout d'abord, représente un problème d'une ampleur sans précédent, compte tenu de l'extrême facilité

et du faible coût de la reproduction d'un logiciel. La richesse inventive déployée pour surmonter, sur un plan technique, des barrières à la reproduction a donné naissance à une activité tout aussi inventive de mise au point d'outils de contournement de ces barrières, commercialisés tout à fait légalement sous le seul prétexte de permettre à l'utilisateur d'effectuer des copies de sauvegardes. La bataille s'est aujourd'hui déplacée du plan technique aux plans du juridique et du pénal et peu de logiciels présentent encore des défenses anti-reproduction. L'acquéreur se voit doté d'un contrat de licence aux conditions duquel le seul fait de réaliser l'installation sur sa machine l'engage explicitement.

Selon les estimations de la *Business Software Alliance* (BSA), une association des plus grands éditeurs de logiciels aux États-Unis, les firmes américaines souffriraient d'un manque à gagner de 2 milliards de dollars par an sur le marché américain et de 12 milliards de dollars à l'échelle mondiale (CQ Researcher, 1993). Ainsi en France, toujours selon BSA, le taux des logiciels piratés atteindrait 51 % en 1995, en régulière régression par rapport aux années précédentes : 57 % en 1994, 66 % en 1993 et 73 % en 1992. Le récent effort judiciaire et d'information, appuyé par les nouvelles dispositions légales en matière de propriété intellectuelle, vise à faire redescendre ce taux au niveau de 25 % aujourd'hui atteint aux États-Unis. Mais l'attention des éditeurs de logiciels se porte de manière plus inquiète encore sur d'autres continents que l'Europe et en particulier sur l'Asie, où la contrefaçon est devenue une pratique courante, voire même une institution. Ici, la palme du piratage revient, en termes relatifs, à la Thaïlande, où le taux de piratage atteignait 97 % au début des années 90 (quand il était de 40 % aux États-Unis). Mais en termes de valeur les grands marchés représentent une préoccupation essentielle, comme la Chine qui a fait l'objet d'une

(16) MERGES, 1996, p. 279

(17) SCOTCHMER, 1991

(18) ABATE, 1994

pression particulière tant au plan diplomatique international, lois de la négociation du GATT (19), qu'au plan privé de la part d'entreprises individuelles, comme Microsoft qui entend réussir sa stratégie de pénétration de l'immense marché chinois (20)

Au-delà de cette question, certes importante, mais qui ne pose pas de problème de protection spécifique dès lors que l'on dispose d'un cadre juridique privatif, un problème plus délicat est celui de l'imitation, dont l'aspect le plus spectaculaire est le phénomène des « clones » logiciels. Ce phénomène est apparu dès lors qu'abandonnant peu à peu un univers cloisonné, le monde de l'informatique s'orientait vers l'élargissement d'espaces de compatibilité (21). Délicate s'il en est, la question de l'imitation va poser un problème difficile au législateur, quant à savoir si quelles bases objectives on peut se fonder pour apprécier la similitude et dans quel cadre juridique on peut sanctionner les abus. Au-delà même de l'aspect technique et juridique, c'est, on le verra, tout le problème de la dynamique de l'innovation qui est ici concerné.

La question centrale quant à la définition d'un « bon » système de propriété intellectuelle est relative à l'étendue et au niveau de la protection, « ce qui devrait ou ne devrait pas être protégé, ce qui est ou n'est pas effectivement protégé » (22). Et sur cette question les points de vue diffèrent selon que l'on se situe dans la communauté technique ou juridique, selon que l'on parle au nom de grands constructeurs de matériels, de grands développeurs de logiciels, de petits éditeurs, des SSII ou des utilisateurs.

Selon l'OTA (23), on peut distinguer quatre composantes constitutives d'un programme informatique : la fonction, le design externe, l'interface utilisateur et le code. Chacun d'entre eux fait l'objet d'un débat spécifique quant au mode approprié de protection.

Dans la législation américaine les différentes composantes de la fonction peuvent faire l'objet de brevets, sous réserve de satisfaction des critères de nouveauté et de non-évidence. Nous avons vu le type de problèmes qui s'ensuit. Les algorithmes mathématiques restent toutefois, à ce jour, exclus du champ de la protection.

Le design externe spécifie les entrées/sorties et les fonctions de communication avec le programme (formats, commandes, protocoles), il est clair que la question de la protection des interfaces reste un des points délicats du dispositif, aussi bien aux États-Unis qu'en Europe.

L'interface utilisateur recouvre les différentes conventions de communication homme-machine : langage de commande, menus, graphiques et icônes. Elle rentre pour l'essentiel dans le champ du *look and feel* (graphisme et économie de l'interface – système de boutons, d'icônes et de menus déroulants par exemple), vis-à-vis duquel la législation reste dans le flou et la jurisprudence est encore balbutiante.

Le code enfin est protégé par le copyright. La question essentielle demeure celle de l'appréhension du niveau de similitude, vis-à-vis duquel on peut dire qu'il y ait ou non contrefaçon. À quel niveau d'abstraction deux programmes peuvent-ils être considérés comme similaires ?

La question de fond, telle qu'elle apparaît très clairement dans le débat américain, demeure quant à l'éventuelle extension de la protection de l'expression (contenant) à l'exprimé (contenu) à savoir, d'une part, le « *look and feel* » d'un programme et, d'autre part, les méthodes et les concepts. « Des arguments ont été avancés pour étendre le copyright des programmes informatiques, en protégeant les méthodes d'analyse utilisées par les programmeurs et pas seulement l'expression « écrite » de leurs idées. (Mais) les procédures et méthodes véritablement mises à l'œuvre dans les programmes

(19) CORREA, 1993

(20) ENGARDIO, 1996

(21) DELAPIERRE et ZIMMERMANN, 1994

(22) OTA, 1992

(23) Idem

n'entrent pas dans le champ de couverture du copyright (24) » Libre-arbitre est laissé aux tribunaux de juger ce qui doit ou ne doit pas être protégé par la loi

La référence en matière de jurisprudence est ici l'affaire *Whelan v Jaslow* en 1986 qui a semblé fonder une extension de la protection du copyright du code littéral exprimant un programme à sa structure, à son enchaînement et à son organisation. Ce jugement a dès lors fait jurisprudence sur la base d'une « *substantial similarity* », qu'elle soit constatée à un niveau d'ensemble ou en distinguant la similarité des idées sous-jacentes ou de celle de l'expression de l'idée sous-jacente

Concernant la question du « *look and feel* », dans l'affaire *Lotus v Paperback* (1987) c'est l'extension de la protection par copyright à la structure, la séquence et l'organisation des menus de commande qui était en cause, « comprenant la conception d'ensemble, le choix des polices, des mots, des symboles utilisés pour représenter chaque commande, la structure et l'ordre des termes de commande et de chaque ligne des menus, la présentation de ces termes de commande à l'écran et des sollicitations de réponse (25) »

Mais le bel édifice juridique forgé au fil du temps s'est vu plus récemment désavoué par la Cour d'Appel de New York, lors d'un jugement rendu le 23 juin 1992 (*Computer Associates v Altai*). Les magistrats se sont très clairement démarqués des jugements rendus antérieurement en estimant « qu'une ressemblance de structure ne constituait pas une violation du droit d'auteur » (26). Ce jugement constituait l'amorce d'une interprétation beaucoup moins stricte de la protection juridique fondée sur le droit d'auteur et devait

contribuer à mettre un frein à la vague débridée de poursuites tous azimuts engagées par certains leaders, soucieux de préserver une position dominante à l'égard de leurs imitateurs ou supposés tels (27)

La législation française, selon la loi de 1985, exclut du champ de la protection les idées et les principes, mais reste floue sur les conséquences de cette exclusion. Poussée à davantage de précision, la Directive Européenne étend cette exclusion à la logique, aux algorithmes et à la langue de programmation. La loi française du 10 mai 1994 effectue un pas en avant supplémentaire sur la question en utilisant le concept de « logiciel », là où les lois plus anciennes et la Directive Européenne elle-même se référaient au concept de « programme d'ordinateur ». Ici la notion de logiciel désigne non seulement une série d'instructions et de spécifications internes, « mais elle intègre aussi la totalité des spécifications externes, c'est-à-dire les dialogues homme-machine, la présentation des écrans, les entrées et sorties, l'ergonomie, voire la mise en scène ou le « *story board* » pour les logiciels de jeux ou les logiciels multimédias » (28)

La question du « *look and feel* » enfin divise grands éditeurs et utilisateurs. Ces derniers souhaitent au contraire bénéficier des externalités (*) que leur procurerait un environnement de similarité, notamment en raison des effets d'apprentissage dont ils pourraient bénéficier. « Nous voulons des logiciels dont le graphisme et l'ergonomie se ressemblent afin de réduire l'apprentissage nécessaire pour faire passer des utilisateurs d'un logiciel à un autre (29) ». Ce sont les effets d'externalités de réseaux au sein du marché du logiciel qui sont en cause ici et qui soulèvent plus généralement la question de la révélation des interfaces nécessaires à la construction de produits logiciels complémentaires

(24) SCHACHTER, 1987

(25) Lotus a obtenu gain de cause aux termes d'un jugement nuancé ; voir pour les détails OTA (1992), p. 73

(26) BRANDT et alii, 1992

(27) C'est le cas par exemple de Lotus contre Botland en 1990 (Quatro Pro, jugement novembre 1992 à Boston). En 1990 Lotus avait obtenu gain de cause dans une affaire similaire contre Paperback Software International selon les mêmes arguments qui ont été rejetés par la Cour de New York dans l'affaire *Computer Associates v Altai*

(28) BENSOUSSAN, 1994

(29) SCHACHTER, 1987

(*) Effets positifs ou négatifs – resp. avantages ou inconvénients – produits pour une situation ou une action économique sans compensation, correspondante en vente sur le marché

Interfaces et décompilation

En avril 1992 à San Francisco, le fabricant de consoles de jeux Sega a obtenu gain de cause contre Accolade, parvenant à contraindre cette dernière à abandonner et à retirer du commerce ses logiciels de jeux compatibles avec les consoles Sega. Ce jugement a constitué une première dans la jurisprudence et il convient d'en préciser la signification. Pour ce faire, Accolade avait dû étudier, par « *reverse engineering* » (analyse d'un objet technique par déconstruction), une copie temporaire du logiciel des consoles Sega (système d'exploitation) afin de déterminer les conditions de compatibilité. Cette copie a été considérée par la Cour de San Francisco comme une atteinte au droit d'auteur de Sega.

Ce sont toutes les actions autour du concept de compatibilité qui sont ici concernées et c'est pourquoi ce jugement a donné lieu aux États-Unis à une véritable mobilisation. On a vu ainsi se constituer deux groupes prenant fait et cause pour chacune des parties. D'un côté, les défenseurs d'Accolade argumentent qu'un tel précédent étoufferait la concurrence et qu'il faut au contraire défendre la croissance et l'esprit d'innovation. De l'autre, IBM et Apple prennent le parti de Sega et défendent les positions acquises des leaders. L'une et l'autre des coalitions représentent les deux termes du paradoxe schumpétérien, entre retour sur investissement pour l'innovateur et dynamique de l'innovation, dans un univers industriel où les externalités de réseau résultant avant tout de la disponibilité d'une offre de produits complémentaires compatibles.

Et ce n'est certainement pas un hasard si lors du débat passionné autour de l'interopérabilité, dans le cadre de la préparation de la Directive européenne, on a trouvé des coalitions strictement similaires. Pour favoriser l'interopérabilité (des ordinateurs et des programmes), le texte européen souhaite contraindre les éditeurs à communiquer les informations nécessaires à la réalisation

d'interfaces et, à défaut, autorise la décompilation pour accéder à ces données.

Certains commentateurs estiment que le texte européen est typique du paradoxe auquel se heurte le législateur. D'un côté il vise à favoriser un élargissement de la concurrence, donc une dynamique industrielle fondée sur la compatibilité et l'interopérabilité. « Mais (de l'autre) lorsqu'une entreprise a, grâce à sa capacité d'innovation et à son marketing, réussi à imposer un standard de fait, elle sera obligée, parce que c'est de l'informatique, de fournir tous les éléments permettant à des concurrents de faire des produits compatibles », les concurrents pourront ainsi développer et proposer des modules complémentaires, voire des programmes beaucoup plus complets à un niveau de rentabilité bien plus élevé (30).

Ici la loi française veut aller plus loin encore et autorise la décompilation des programmes et leur analyse en vue d'assurer l'interopérabilité. Mais à une volonté claire se heurte une réalité plus complexe qui rend délicate l'appréciation du champ de la décompilation, nécessaire à la réalisation des interfaces, et celle de la frontière entre, d'un côté, une analyse destinée à assurer l'interopérabilité et, de l'autre, l'accès à des informations qui équivaldraient à une violation du droit d'auteur. Elle constitue néanmoins une incitation pour les développeurs à publier les informations relatives aux interfaces, à choisir clairement des standards d'interopérabilité et à répondre aux demandes individuelles des utilisateurs (31). En ce sens et malgré les difficultés d'interprétation qu'elle soulève, elle constitue une avancée dans une logique de standardisation et de systèmes ouverts.

La standardisation comme clef de dépassement du dilemme schumpétérien ?

« Les externalités de réseau soulèvent le problème de la coordination entre agents décentralisés [...] Le système juridique de protection peut jouer un rôle défavorable

(30) BENSOUSSAN, 1991

(31) VAN DORSSSELAERE, 1994

quand il n'est pas bien ajusté au fonctionnement du marché. En concevant une protection juridique des œuvres intellectuelles qui comporte des externalités de réseau, les législateurs contribueraient à renforcer les avantages de la standardisation (plus grande disponibilité des produits, abaissement des coûts de recherche, accroissement de la concurrence) tout en minimisant les contreparties négatives qu'un libre accès aux nouvelles technologies comporte quant aux incitations à innover (32) »

Dans un contexte à fortes externalités de réseau, un système de protection doit viser à la fois l'incitation à d'utiles informations et en même temps la pleine réalisation des bénéfices issus des effets de réseau. En d'autres termes, ce qui doit être protégé pour le logiciel, ce sont les innovations effectives, mais non les interfaces-utilisateur, formats de stockage de données et autres aspects relativement arbitraires mais indispensables au jeu des innovations de complémentarité, sur une base de compatibilité.

« La loi sur le copyright protège les éléments « arbitraires » ou « expressifs » d'une œuvre, mais ne protège pas spécifiquement les idées utiles qu'elle comporte [] Des innovations d'une réelle utilité peuvent ainsi être sous-protégées alors que certains éléments arbitraires, l'interface utilisateur par exemple, seront couverts et pourront générer une rente importante si ils deviennent des standards de fait (33) ». Et ces aspects arbitraires, à partir du moment où ils sont adoptés, acquièrent une valeur sociale, qu'ils n'avaient pas en eux-mêmes. Il n'y a par conséquent aucune raison économique que l'innovateur perçoive une quelconque rente pour cet aspect purement arbitraire et non inventif.

Dès lors, si l'on veut bien considérer que la question de l'interopérabilité équivaut à un problème de coordination entre agents décen-

tralisés, à travers la mise en place d'un espace de compatibilité, alors les spécifications d'interfaces devraient être vues sous l'angle de leur nature de biens publics. Elles permettent d'assurer la conciliation entre l'efficacité individuelle des agents et l'efficacité sociale du « réseau d'innovation » qui en résulte. « L'entrée sur le marché, la compétition et l'innovation deviennent plus faciles si les concurrents potentiels peuvent se contenter d'améliorer un seul des éléments du système, dynamisant ainsi le marché de tels composants, plutôt que d'être obligé d'innover sur l'ensemble d'un « système » (34) »

« Cette fois la normalisation n'est pas réductrice, mais facilitatrice de variété, par la possibilité qu'elle offre d'une mise en œuvre de complémentarités technico-productives. Bien plus, elle engendre une accélération du progrès technique en s'appuyant sur des effets de « lock-in » (effet de cliquet ou irréversibilité) qui l'accompagnent par effet de retour des rendements croissants d'adoption dont elle ouvre la voie (*) C'est en somme le rôle même et le statut du standard qui s'est profondément transformé. Jusqu'alors outil de hiérarchisation au sein de structures verticales, il est ainsi devenu vecteur d'harmonisation au sein des structures de nature résiliante qu'engendrent ces complémentarités (35) »

Il est clair que la production de normes, en ce qu'elle permet l'interopérabilité, apparaît comme une voie de réponse à certaines limites de la protection des logiciels. La normalisation ne peut pas cependant être vue comme une alternative stricte à la protection. La généralisation de l'adoption d'un standard n'est pas nécessairement incompatible avec la propriété intellectuelle. Elle peut au contraire la pérenniser en offrant des solutions de non-enfermement, qui à la fois permettent à l'innovateur de bénéficier d'effets d'avance et préservent les possibilités d'effets de complémentarité, renforcés

(32) MENETI L., 1987

(33) FARRELL, 1989

(34) Idem

(35) DELAPIERRE et ZIMMERMANN, 1993

(*) L'intérêt économique devenant de plus en plus important au fur et à mesure de l'augmentation du nombre d'utilisateurs

par le caractère fortement systémique du produit et donc de l'industrie informatique

C'est là le fondement de la notion de *système ouvert*, qui est au centre des concepts contemporains les plus avancés en matière de normalisation et qui consiste à fournir des standards non-propiétaires spécifiant comment les composants interagissent à leur interface. Le logiciel n'est pas en soi objet de normalisation, ce qui doit être normalisé, ce sont les fonctions rendues nécessaires pour des problèmes de communication et de portabilité. À l'opposé, la protection des interfaces confinerait dans des espaces incompatibles. La variété laisserait alors place soit à la position dominante d'un produit, standard *de facto*, soit à la redondance d'une pluralité de produits incompatibles.

* *
*

Traditionnellement, on présente la protection comme nécessaire à la réalisation d'un retour sur investissement pour l'innovateur, donc comme à l'origine d'une incitation à la R & D et à l'innovation. Au contraire, l'absence de protection constitue un frein à l'innovation. Dans un domaine industriel comme le logiciel, caractérisé par une forte dynamique de l'innovation, l'absence d'un système satisfaisant de protection peut au contraire se traduire par l'effet inverse. La quasi-rente engendrée par l'avance temporelle dont bénéficie l'innovateur dépend de sa capacité à en valoriser les effets

L'absence de protection efficace peut dès lors nécessiter, pour l'innovateur, de réduire significativement la durée de vie de ses produits, afin de décourager les stratégies d'imitation et, par conséquent, se traduire par une accélération de la dynamique de l'innovation. Dans un cas comme dans l'autre, l'efficacité sociale de la vitesse de l'innovation est vraisemblablement sous-optimale.

D'un autre côté, les deux termes du dilemme schumpétérien traduisent une divergence de points de vue entre firme et industrie, une opposition entre performance individuelle de l'entreprise innovatrice et efficacité collective de la structure industrielle. Et le paradoxe réside précisément dans le caractère récursif des deux termes. L'efficacité de l'industrie est conditionnée par les performances industrielles et innovatrices des entreprises. Quant à la rente espérée par une firme en relation avec une innovation, elle résulte d'un produit des marges unitaires et du volume de marché, donc d'un arbitrage entre prix et échelle de diffusion.

Si donc la protection de la propriété intellectuelle renvoie avant tout à l'intérêt individuel de l'entreprise, la standardisation apparaît quant à elle un instrument de régulation de l'industrie. À partir du moment où la standardisation, à l'exemple de la problématique des systèmes ouverts, vise la compatibilité tout en respectant l'existence de boîtes noires, elle permet de concilier l'élaboration de designs propriétaires, donc la propriété intellectuelle, et la variété, avec la pleine réalisation d'une innovation de complémentarité et d'externalités de réseau.

RÉFÉRENCES

- ABATE T (1994), « Propriété Intellectuelle – Logiciels cherchent protection sûre », *San Francisco Examiner*, traduit par *Courrier International*, n° 172, 17 février
- BENSOUSSAN A (1991), « Directive européenne gate aux effets pervers ! », *Le Monde de l'Informatique*, 27 mai
(1994), « Droit du logiciel la 1 évolution », *01-Informatique*, n° 1314, 17 juin
- BERTRAND A (1994), « *La protection des logiciels* », Que Sais-je ?, PUF, Paris
- BRANDT R, SCHWARTZ E I, GALEN M (1992), « Propriété des logiciels les verrous sautent », *Business Week/Courrier International*, n° 91, 30 juillet
- CORREA C M (1993), « Legal protection and innovation in the software industry », *World Competition*, vol 17, n° 1, Geneva, September
- CQ Researcher (1993), « Software Piracy Can government help stop the drain on profits ? », May 21, vol 3, n° 19, 433-456, *Congressional Quarterly Inc*, Washington DC
- DELAPIERRE M et ZIMMERMANN J B (1993), « The computer industry, from scale to network effects », in M HUMBERT (Ed), « *The Impact of Globalisation on Europe's Firms and Industries* », Pinter Publ
(1994), « La globalisation, une remise en perspective des structures techniques de l'industrie », *Terminal*, n° 66, hiver
- ENGARDIO P (1996), « Microsoft's Long March », *Business Week*, june 24
- FARRELL J (1989), « Standardization and intellectual property », *Journal of Metrics*, Fall
- KAHIN B (1990), « The software patent crisis », *Technology Review*, April
- LUCAS A (1987), « Le droit de l'informatique », *Thémis*, PUF
- MENELL P S (1987), « Tailoring legal protection for computer software », *Stanford Law Review*, vol 39, July
- MERGES R P (1996), « A comparative look at property rights and software industry », in Mowery D C (Ed), « *The International Computer Software Industry* », Oxford University Press
- OTA (1992), « *Finding a Balance, Computer software, intellectual property and the challenge for technological change* », OTA-TCT-527, Washington DC
- SCHACHTER E R (1987), « Software protection in the throes of legal morass », *Datamation*, June 1
- SCOTCHMER S (1991), « Standing on the shoulders of giants – cumulative research and the patent law », *Journal of Economic Perspectives*, vol 5, number 1, Winter, 29-41
- VAN DORSSELAERE B (1987), « Protection des logiciels le brevet est possible », *Le Monde de l'Informatique*, 28 septembre
(1994), « La décompilation, une incitation à la publication », *01 Informatique*, n° 1314, 17 juin
- YAMAMOTO K (1989), « Japan's software industry », *Japan Computer Quarterly*