

コンピュータ・プログラムの著作権法保護と社会のネットワーク化

辰巳直彦

神戸学院大学法学部

本来技術としてのコンピュータ・プログラムにつき、著作権法により保護する場合には考慮されなければならない点が多岐にわたる。このうち3つの視点を取り上げた後、コンピュータ・プログラムの保護範囲を、特にプログラムに含まれるアルゴリズム、インターフェース情報、通信プロトコルとの関連で述べる。次いで、コンピュータ・プログラムのリバース・エンジニアリングが著作権法のもとで許容されるべきこと、及びリバース・エンジニアリングの結果得られたアイデアたる技術情報を利用して機能的に別の又は互換のプログラムを開発することの許容性とその際の問題点を指摘する。

Copyright Protection of Computer Programs in the Context of Network Society

Naohiko Tatsumi

Kobe Gakuin University, Faculty of Law

518 Ikawadani-cho Arise, Nishiku Kobe-shi, Hyogo 673, Japan

The author suggests three fundamental points which must be taken in account in discussing the copyright protection of computer programs. On the basis of such points he analyzes the scope of protection of computer programs with specific reference to algorithm, interface specifications and communication protocols. He also discusses the permissibility of reverse engineering of computer programs and development of functionally different or compatible programs based on the technological information obtained from reverse engineering.

1. はじめに

今日、コンピュータによる情報処理技術と情報通信技術とが飛躍的に発展しつつある。そして、これにともない、これらを社会のインフラストラクチャーとする社会の高度情報化が叫ばれるとともに、広範な情報関連市場が提供されつつある。こうした中で、コンピュータ・システムにおいて情報処理や通信を司るコンピュータ・プログラムの重要性が認識されつつあるとともに、その模倣の容易さの故に安易なただ乗りを誘発する傾向があることは否めない。そこでこれに対処すべく、先進国の中では米国が1980年の著作権法改正とともにコンピュータ・プログラムを明示的に保護の対象としたことに続いて、各国でも著作権法によりコンピュータ・プログラムの保護を図っていくことが、大勢となっている。わが国においては、昭和47年頃よりプログラムを経済財とみる通産省の独自立法 (*sui generis*) による保護の主張と、文化庁の著作権法による保護の主張とがぶつかるような形で、プログラム保護のあり方について激しい議論が展開されたが、結局は米国からの圧力もあって、文化庁の意向に沿うような形で昭和60年の著作権法一部改正 (昭和60年法律第62号) により、コンピュータ・プログラムが著作権法によって保護されることが立法上明確化された。

2. プログラム著作権を考える基本的視点

このようにわが国においては、著作権法の昭和60年改正により、コンピュータ・プログラムを「電子計算機を機能されて一の結果を得ることができるようにこれに対する指令を組み合わせたものとして表現したものをいう」と定義し (同法2条1項10号の2)、さらに「プログラムの著作物」 (同法10条1項9号) という新たな著作物の類型を設けており、コンピュータ・プログラムが著作権法により保護されることについては全く疑いを挟む余地はない。しかも、著作権法の一般原則としては、①表現/アイデア (ないしは形式/内容) という二分法 (*dicothomy*) に立ち、著作物の表現 (形式) を保護し、そのアイデア (内容) を保護するものではないこと、②著作権法による保護は独立創作を排除せず、独立して創作された二つの表現がたまたま同一であっても、その何れか一方が他方を排除して、その保護が否定されることはないこと、さらに③アイデアを表現する手段に限りがある場合には、表現とアイデアの一致すなわち混同 (マージ; *merge*) があり、このときにその表現に保護を与えることはとりもなおさずアイデアの独占を与える結果になるので保護が与えられないこと (マージャー理論; *merger doctrine*) — これらは、特に米国において主張されていることであるけれども、わが国においても一般に承認されるところでもあろう。こうして、コンピュータ・プログラムは著作権法及びその一般原則に基づき保護されることになったのであるが、その適正な保護を考える際には、開発者の利益保護はもちろんのこと、さらには開発者間の適正な競争、コンピュータ・システムのユーザの利益、及び社会一般の技術の発展に対する要請を考え、次のような基本的視点が念頭に置かれるべきであると考えられる。

まず第一点として、コンピュータ・プログラムは、コンピュータに対する指令をコンピュータ言語で記述する「表現」としての側面があるとともに、そこに記述された処理の手順に従ってコンピュータを稼働させるという「機能」を有しているということである。実際プログラムは、コンピュータ内で稼働している時には、電気的には回路の開閉の状態を表しているにすぎないのであり、またそれ自体技術製品であることから、一面ではハイブリッドな機能作品 (*work of function*) といわれる所以である。従って、こうしたコンピュータ・プログラムにあっては、必然的に表現とアイデアの区分はあいまいとなり、その結果これに余り広範な保護を付与することになると、それに応じて広範な機能及びその機能を実現する技術的手法の保護に至りかねず、また、これがために著作権法がプログラムの先行開発者に有利に働き、その者が類似プログラムを開発した後発者に対して権利侵害の主張を申し立て、この後発者を事実上市場より排除し、プログラムに含まれる技術的手法を独占するための武器を与えうる危険性があることには注意を払うべきである。そこで、コンピュータ・プログラムを著作権法という主としてコピーから著作者の利益を保護する法制により保護することになった根拠としては、プログラムの開発に多大な知的労力と投資がかかる反面では、他の技術とは異なり模倣やただ乗りが容易になされうということに求め、且つこれを指針として適切な保護範囲の探求がなされるべきである。

次に第二点として、コンピュータ・プログラムが従来の著作物と大きく異なるのは、その本質が「技術」にあ

り、経済財・産業的所産としての性質を有するという点である。この点、従来の小説、音楽や絵画等の著作物が「多様性」の追求の上に成り立っているため（伝統的著作物の連心性）、これらについて広い保護範囲を設定しても特にならぬ障害が生じることもなかった。これに対し「技術」の特性としては、まず①その性質上、既存の先人の積み重ねに関わる社会共通の技術的基盤に立脚して、個々の技術者が付加的な貢献をつけ加え、これにより社会の技術的共通基盤のより一層の拡大と、より高いレベルへの移行を経て、累積的に発展するものであり（技術の累積的発展性）、且つ、②「多様性」よりもむしろ「合理性」の追求を通じて、より良い技術へと収斂する傾向が強いことである（技術の球心性）。従って、こうした技術の産物に対して容易に独占的かつ広範な保護を与え、類似生の故に安易に侵害を認定し、また技術の進歩と発展の不可欠なプログラムに含まれている技術情報に対しアクセスを認めないような法律解釈をするということになると、技術競争を通じた技術の発展と進歩を阻害し、ひいては社会一般がより良い技術の恩恵に浴することを妨げる結果となる。そこで、このような観点からもプログラムの適正な保護範囲、さらにはいわゆるプログラムに含まれたアイデアたる技術情報を抽出するリバース・エンジニアリングの許否をも考えるべきであるとする。

さらに第三点としては、情報処理技術と情報通信技術の結合により進行しつつある高度情報化社会にあつては、コンピュータは単体としてではなく複雑なネットワークを含むシステムとして現われ、その中にあつてコンピュータ・システムを構成するハードウェア及びソフトウェア等の諸要素の間においては、互換性（compatibility）、相互接続性（interconnection）が保証され、これを通じて次のようなユーザの利便性が確保されることが要請されるといえる。すなわち、①ユーザが用途に合わせて最新・最適のハードウェアを選択できること、②かつユーザが異なるメーカーの機種上でも多くの同一のプログラムを走らすことができ、このことがプログラムを供給する側に一層の競争効果を生じさせ、またその跳ね返りとしてユーザが一層安価で最適のプログラムをより広範な範囲から選択できるようになること、③ユーザが以前のもとは異なるメーカーのシステムに乗り変えた時でも、既存のユーザ資産を受け継ぐことが出来ること、④複数の異なるベンダー、すなわちマルチベンダーによるコンピュータ・システムやネットワークを構築し、しかもシステムの広域化に対処できること。ただこれらは将来的にはコンピュータ・システムの構成要素間のインターフェースやシステム間どうしの通信に関わる技術仕様や規約の「規格化」ないしは「標準化」（standardization）によりある程度達成されることが期待され、実際、通信プロトコルについてはIOS/CCITTによるOSI（開放型システム間相互接続）モデルによる標準化の努力がなされている。しかし他方では、一般的にこうした仕様の優劣は、現在差し当たっては自由競争による改良と淘汰に委ねられるべき点も多くあることも否定できず、そうした中で、当面システム構成要素間の互換性や相互接続性等がユーザには保証されない状況が多々存在することが考えられる。しかし、その場合であっても局部的に互換性又は相互接続性を達成するための手だてを法的に保証することが、プログラムの適正な法的保護を議論するに当たっても考慮されるべきであり、且つそうすることがコンピュータ関連市場の競争を刺激し、また標準化を促す要因たりうるものとする。そしてこの観点からもプログラムのリバース・エンジニアリングの許否、及びその結果に基づく互換プログラムや相互接続を可能にするプログラムの開発の是非を考えるべきであると思われる（むしろ許容されるように考えていくべきである）。

3. プログラムの保護範囲

そこで、前項で述べた諸視点を念頭において、プログラムにおいて著作権法により保護される表現とは如何なる範囲又は抽象度のものをいうのかについて考えてみる。この点、周知の如く米国では、1985年頃を境としてコードの文字通りのコピー（literal copying）を越えた事件が裁判上取り扱われるようになり、その先例的判例がプログラムのアイデアを、その果たす抽象的な目的又は機能にあると捉えた上で、それに不可欠でないものは全て表現に該当するとして、プログラムの全体的な構造（structure）、流れ（sequence）又は構成（organization）も広くプログラムの表現として保護の対象たりうるものと述べるに及んで多くの議論を呼んでいる（Whelan Associates, Inc. v. Jaslow Denatal Laboratory, Inc., 797 F.2d. 1222 (3 Cir 1986)）。これに対して、わが国では、最近の東京高裁の仮処分事件において、申立人指摘の類似部分につき、指令の組み合わせはハードウェア

アの規制により同様のものにならざるを得ないこと、または極めて一般的な指令の組み合わせをしているという理由で申立人のプログラムの創作性を否定した上で、プログラムにおける「処理の流れ」自体は著作権法10条3項2号に規定されている「解法」であって著作物として保護されないと述べ、申立人の主張を排斥した事例がある（東京高裁決定平1.6.20判時1322号138頁）。この判決のいう「処理の流れ」が、米国の判例にいう「流れ」(sequence)に該当するものであるなら対立した立場が表明されたことになる。もっとも米国においても先の立場が確定したとはいえず、またわが国においても、コンピュータ・プログラムが著作権法によって保護されることになった結果、理論的にはプログラムの複製のほか翻案についての権利をプログラムの著作者が専有することになったが（著作権法27条）、その場合に、先の判例にもかかわらず、具体的に外面的な表現の類似性がなくても表現内容が類似していたり、言語変換のなされたときに、いかなる基準を満たすときに翻案権の侵害が成立し得るのか、特に著作権法がプログラムの著作物について保護の及ばないものとして「プログラム言語」のほか「規約」及び「解法」を明文上挙げている（著作権法10条3項）こととの関係で、解釈上依然議論の余地を多く残している。

もっともわが国では、コンピュータ・プログラムにおける問題解決のための処理の手順は一般にはアルゴリズムと言われ、これはアイデアであり、著作権法上「解法」に該当するものとして保護されないとされているが、他方、プログラムというものは、そもそもコンピュータにおける処理の手順をプログラム言語で記述したものであることを考えると、わが国ではプログラムは、その外面的表現形式に即してその同一性の範囲を画することを意図していると思われる。他方、わが国の著作権法は「翻案」という概念を認めており、これは著作物の表現に含まれる内面的形式を維持しながらその外面的形式に変更を加えることであるとされている。そしてこの「翻案」がコンピュータ・プログラムの場合にどの範囲まで及ぶかという問題は、結局コンピュータ・プログラムの表現の同一性の範囲の問題と表裏一体の関係にあり、またわが国では先の保護されないと規定されている「解法」との関連を考慮すれば、コンピュータ・プログラムにあっては、理論的には外面的形式にかなり近接したところに内面的形式が認められることになり、その結果、翻案の範囲はかなり限定され、従って学術的な著作物一般についていえることでもあるが、基本的な筋・仕組みまで保護する小説の場合等と比較すると、プログラムの著作者の翻案権の及ぶ範囲は比較的狭いものと解されているように思われる。

ところでこうした見解は、その本質が技術であるコンピュータ・プログラムを著作権法により保護する理由を、プログラム開発には多大な知的労力と投資がかかる反面、他の技術と異なり容易にコピーやただ乗りがなされ得ることに求め、累積的に発展していく技術の特性を鑑み、プログラム開発者の利益と技術に対する社会一般の利益とのバランスを図ることを指針とする限りは、大凡正しい方向を示すものであると考える。しかし、一方ではプログラムというものは、そもそもコンピュータ内における処理の手順をステップ毎にアルゴリズムとして記述するものであるのにも拘らず、プログラムにおけるいかに詳細なアルゴリズムであっても、これを一律に全て著作権法により保護されないアイデアとして著作権法10条3項3号にいう「解法」に該当するものとする、万人が自由にこれを利用し、これにより必然的にもとのプログラムと同一の指令の組み合わせからなる表現のプログラムの作成を認めることに等しい結果になる。これでは全てのプログラムについて表現とアイデアとが一致ないしは混同（マージ）することを認めることと同じになり、結局はプログラムの著作権法による保護を否定してしまうことになる。ただ、こうした矛盾はハイブリッドな機能作品としてのプログラムの特性から由来するものであると考えるが、このことはプログラムの法的保護を考えるにおいては、何らかの程度アルゴリズムを考慮に入れざるえないことを示唆するものでもある。というのも、実際のところコンピュータ・プログラムにおいて本質的なのはアルゴリズムであって、それがコンピュータ言語で表現されていようが、その他の形態で表現されていようが、結局はそれをコンピュータに理解可能な形態に翻訳・解釈するコンパイラないしはインタープリターがあるかないかの違いだけなのであり、それさえあれば如何なる形態のものでもコンピュータに対する指令を与えるものとしてコンピュータ・プログラムといえるのである。従ってコンピュータ・プログラムの法的保護を考えるに当たっては、確かに現在ではコンピュータ・プログラムがプログラム言語で表記されるという現実

は無視できず、またそれが著作権法により保護されることになった結果として、著作権法の「表現」を保護するという原則を尊重しなければならないことは当然であるが、結局はその著作権法も翻案という概念を認めることにより実質的にはある程度アイデアをも保護するに至っている実状に鑑み、且つ冒頭に述べたような諸視点をも視野に入れて、いくらか政策的にコンピュータ・プログラムの表現の範囲及び保護範囲を決定せざるをえないものと思われる。

そこでプログラムにおけるアルゴリズムについていえば、一面では、これは問題解決のための処理の手順であり、それ自体としてはプログラムのアイデアであることは否定できない。しかし他面では、プログラム開発過程において、一定の固有の問題を解決するため、既存の基本的なアルゴリズムや新たに発見したアルゴリズムを幾多も選択・配列し、より複雑なアルゴリズムを構成しながら、その精緻化を図っていく過程が、ソフトウェア・エンジニアやプログラマーの最も多大な知的労働と個性の発揮される部分であり、またこのようにしてアルゴリズムが十分詳細に精緻化された時点では、それが創作的な指令の組み合わせの表現としてのコンピュータ・プログラムの外面的形式に大きな影響を及ぼしうことは看過できないものと思われる。それ故、最終プログラムに表現されている限りにおいて、十分詳細に精緻化されたアルゴリズムは、プログラムの創造性やその指令の組み合わせの外面的形式を規定するという意味で、著作権法上考慮しうるに足る特性を有しているものといえる。

従ってコンピュータ・プログラムにつき、著作権法による保護を考える際には、プログラムの外面的形式から出発することが正しいとしても、平均的プログラマーから見てプログラムの指令の組み合わせの外面的形式の本質的部分を規定している限りにおいては、プログラムから読み取れる十分詳細なアルゴリズムは、実質的にプログラムの表現を構成するもの（内面的表現形式）と解しても差し支えないのではないかと考える。そこで例えば、他社開発のプログラムにおけるこうした十分詳細なアルゴリズムを利用してプログラムを作成し又は言語転換することは、外面的なステップ毎のコードの類似性がなくとも、実質的にもとのプログラムの指令の組み合わせを利用するものとして「翻案」に該当するものと考えらるべきであろう。さらに、あるプログラムからその十分詳細なアルゴリズムを記述したフローチャートを作成した場合に、そのフローチャートが平均的プログラマーから見て、もとのプログラムの指令の組み合わせの本質的部分について特定できる程のかなり詳細なものならば、これを基に別のプログラムを作成するという程度では、同様の理由で著作権法上「翻案」に該当しよう。これに対し、プログラムにおける十分詳細なアルゴリズムとはいえない、より抽象的なプログラムの構造、処理の流れや手順（これも技術的にアルゴリズムであるが）は、まさに著作権法10条3項3号の「解法」に該当するものとして保護されず、万人が自由に利用することができるものと考えらる。そしてこのあたりが政策的にみても、先行プログラムのただ乗りを防止するのに必要な境界であり、プログラム開発者の利益とプログラムに含まれる技術についての社会的利益とのバランスを図る接点であるように思われる。

次に、わが国の著作権法10条3項2号によれば、プログラムの保護はプログラムにおける「規約」に及ばないとされている。この規約とはインターフェースやプロトコルといわれるものを指し、その中でも異なった複数のプログラムあるいはプログラムとハードウェアとが互いに制御の受け渡しをしながら連動し作動するための規則（プログラム間インターフェース、ソフトウェア／ハードウェア間インターフェース）、および距離的に離れた異なった情報機器内のプログラム（通信プログラム）が互いにデータの送受信を行いうるための規則（プロトコル、通信規約）で、しかも特定のプログラムに含まれているものを意味するものと考えられる。これらは、互換機の製造やコンピュータ間の通信にとっては不可欠な技術仕様ともいえる情報であり、もしこれに独占権を与えるとすると、互換機の製造や周辺機器の他社による製造を不可能にし、さらには異機種間の相互接続や通信がなしえなくなるために、特定ベンダーのコンピュータ・システムを利用するユーザは長期にわたって同一ベンダーのシステムにロック・イン（lock-in; 拘束）されることになる。また、そのベンダーが市場で独占的な地位を占めている時にはその独占は一層助長され、競争が阻害されることにもなる。従って、こうした情報それ自体にはコンピュータ・システムの構成諸要素間における互換性や相互接続性を確保する社会的利益の必要上、及びこうしたものもそれ自体としては一般にアイデアに属するものとして保護が及ばないとするのが著作権法の立場と

考えるべきである。ただそれ以上に、インターフェース又はプロトコル等の規約が特定のプログラムにおいて実現されている場合に、一般論としてそのプログラムは著作権法上表現として保護されるといえるが、そのプログラムの依拠したインターフェース又はプロトコルの仕様が、個々の一定のプログラムにおける指令の組み合わせを実質的に規定してしまっていると見られる場合もあろう。そのときに、こうした規約を実現し且つ互換性又は相互接続性を達成しうる別のプログラムを作成しようとすれば、そのプログラムのこうした規約を実現する部分については、不可避的にもとのプログラムと表現が同一になったり又実質的に類似せざるを得ないことになる。そこでこのような時には、もとのプログラムの形態における規約を表現した部分には保護が及ばないとするのが著作権法の立場であると考えべきである。しかし、どのような場合にこうした結果となるのかは、個々に検討して決すべきであらう。なお、インターフェースやプロトコルの開発には多大な費用と時間とを必要とし、最初にこれを開発した者に何らかの利益を与えることなく、これを万人が自由に無償で利用してもよいとすると、最初に開発した者だけが不利益を被りセカンド・ランナー以下が利益を受けることになり、不当であるばかりでなく開発意欲を減退せしめるので、独占権はともかく立法論としては対価（報酬）徴収権を認めることが望ましいとする見解がある。しかし、標準化のなされていない現在、より良いインターフェース又はプロトコルの開発は、結果的に、その良さの程度に応じコンピュータ機器や通信関連市場において採用されることになれば、それによる市場支配を通じそれ相応の開発投資の回収と開発意欲の維持を確保できるものと思われるので、私見としては、インターフェース又はプロトコルそれ自体には、差し当たっていかなる権利性を認める必要はないと考えたい。

4. プログラムに含まれる技術情報に対するアクセス

ところでコンピュータ・プログラムは技術製品であり、従ってそれが前提としている技術内容を知りうる手だてを保証することは、市場での先行者の技術を後発者が習得し、自らの技術力を向上させ、先行者と有効に競争することを許し、ひいては社会全般の技術・産業の発展を促すために不可欠のことである。またコンピュータ・システムの構成要素間の互換性や相互接続性を確保するために他社開発のプログラムを調査解析し、そこに含まれているインターフェースやプロトコルの仕様を知る社会的・技術的要請もある。なんとすれば、互換性や相互接続性の確保それ自体は、なんら非難されるべき筋合いのものではなく、むしろ社会におけるコンピュータのネットワーク化とコンピュータ・システムのマルチベンダー化を背景に、製品競争を促進するという観点からは望ましいといえる。そこでコンピュータ・プログラムに含まれるアイデアである技術情報を抽出するための調査解析行為である所詮リバース・エンジニアリングを、著作権法は明文をもってこれを許容していないために、これをいかに考えるかが問題になる。ただ、コンピュータ・プログラムに含まれる技術内容を抽出するためには、目的のプログラムを実行させたり、プログラム付属のマニュアルを読んだり、さらにはテスト・データを使って通信回線のトレースを行ったり、またテスト・プログラムを目的プログラムとともに実行させたり等して、ある程度の情報は入手でき、しかもこうした行為は、通常、プログラム著作権者の排他的権利の内容に該当する行為を伴わないといえるので、その限りでは、このような態様のリバース・エンジニアリングはそもそも著作権法の関知するところではないといえよう。しかしながら、より詳細で正確な情報を必要とする時には、適正な料金によるプログラム権利者の情報開示がない限りは困難な問題に直面する。すなわち、こうした場合、コンピュータ・プログラムは一般の著作物とは異なって、通常、人が読み理解することの出来ない0と1の羅列であるオブジェクト・コードの形態で、且つ磁気テープやフロッピー等の媒体に乗せられて市場に出されているために、プログラムに含まれている技術内容を知ろうすれば、プログラムをコンピュータ内のメモリー（主記憶装置）にロードした上で、プログラムのダンプ（特にプリントアウト）をしたり、また逆アセンブル又は逆コンパイルによりソース・コードに転換しなければならないことになる。ところがこうした行為が著作権法上の複製、翻訳又は翻案に該当するものとする、著作権者の権利を重視する余り競争を通じた技術や産業の発展を阻害しかねず、また高度情報化社会の要請に答えることも出来なくなる。そこで、他の典型的な技術保護法制である特許法（69条第1項）や実用新案法（26条）あるいは半導体集積回路の回路配置に関する法律（12条3項）が、技術保護の

要請と技術発展の要請とのバランスを図りつつ、権利者の排他的権利に該当するような行為を伴うリバース・エンジニアリングをも一般に認めていることとの関係で、著作権法上同様なリバース・エンジニアリングが許容されるか否かが問題になる。

これについてわが国において、米国著作権法の公正使用 (fair use) の概念に準じ又はこの要件を参考にして、プログラムのダンプ、逆アセンブル又は逆コンパイル等のように、プログラムの複製あるいは翻訳・翻案に当たる行為を伴うリバース・エンジニアリングであっても許容されるとする見解もみられるが、必ずしも一般には明らかではない。そこで、現行の著作権法の解釈論としては、こうしたリバース・エンジニアリングをも許容する方向で、次のように考えるべきであろう。まず、①著作権法は、その1条の目的規定において明示されているごとく、その最終目標は著作者の権利保護と同時に、今や産業の所産であるプログラムを含む著作物の公正な利用にも留意した、文化一般の発展に寄与することにあること、②そこでいう文化には、今日では広く文化的現象すなわち生活形式の総体を意味し、従って広く技術・産業文化も含む趣旨であること、③著作権とはいえ一般の私権であり、その限りでは私権の基本的関係を規律する民法、特にその1条の一般規定により、公共の福祉に違ひ、その権利濫用は許されず、信義誠実にこれを行使用する義務を伴うものであること、④コンピュータ・プログラムを保護することの必要性は強く認められるが、それと同時に高度情報化社会におけるソフトウェア技術の普及と発展は同程度に要請されること、⑤このソフトウェア技術の普及と発展という要請は、まさに高度情報化時代における公共の福祉の内容に合致し、それは翻って著作権法にいう文化の発展に寄与する要請でもあること、⑥リバース・エンジニアリングは、こうした高度情報化社会におけるソフトウェア技術の普及と発展の要請を実現するために必要不可欠な手段であり、その意味で著作物の公正な利用といえるものだが、これをプログラム著作権者がいかなる場合でも理由なく禁止しようとすれば、こうした社会的要請が実現される道が閉ざされ、これがひいては技術・産業文化の発展を阻害し、公共の福祉にも反する結果を来すことになること、しかも⑦リバース・エンジニアリングそのものは究極的には、著作権法10条3項2号乃至3号に掲げられた著作権法により保護されないアイデアたる技術情報を抽出するためになされ、これを禁じることになれば実質的に著作権法は完全なアイデア保護法になってしまうこと、以上の解釈論的論拠より、⑧わが国においてはリバース・エンジニアリングは一般に、たとえ複製や翻案などプログラム著作権者の排他的権利に該当する行為を形式的に伴うものであっても、それは著作物の「公正な利用」に該当し(著作権法1条)、著作権者がこれを著作権の侵害— ないしは著作者が著作人格権の侵害— として主張することは「権利濫用」(民法1条)に該当すると解釈することで、許容されるべきものと考え。ただし、リバース・エンジニアリングの過程で作成される調査対象プログラムの複製物・翻案物(例えばソース・プログラムのプリントアウト)については、目的外に使用し、これを公衆に提示したり、頒布したり等してはならないと解すべきであろう。

こうして、プログラムに含まれる技術情報を取得する手段としてのリバース・エンジニアリングは許容されるべきであると考え。しかし勿論、さらには権利者との契約による方法、またインターフェース情報や通信プロトコル等の不開示が優越的地位の濫用になる場合には独禁法による方法なども考えられるであろう。結局、プログラムに含まれる技術内容や技術情報の取得のためには、なるべく多数の道をオープンにしておき、あとは情報を必要とする者が、情報入手のために取り得る諸方法について効率性やコスト等を考慮した上でなす経済的選択に委ねることが、技術保護と技術競争による技術発展、さらには技術取引とのバランスを図ることに寄与することになるのではないかと考える。ただ、リバース・エンジニアリングの許容性については、先に示したように現行著作権法の解釈として認められ得ると解釈出来ないことはないが、現行のままではどうしても不明確なところが残らざるを得ないので、リバース・エンジニアリングの許容とその限界について必要な争いを避けるためにも立法的措置が取られることが望ましいと考える。

5. フォーワード・プログラミング

他社開発のプログラムをリバース・エンジニアリングした結果取得した技術的アイデア又は情報で、著作権法により保護されないもの— 例えば解法、インターフェース情報や通信プロトコル— は、同機能又は別機能の

プログラムを開発するために自由に利用できることは当然である。これをリバース・エンジニアリングとは区別してフォワード・プログラミング(forward programming)と称することが出来よう。リバース・エンジニアリングそのものは、一般にさまざまな目的のためになされるが、リバース・エンジニアリングはフォワード・プログラミングと結合することによって、機能的技術的に改良されたプログラム開発に寄与し得るという意味において重要である。もっともフォワード・プログラミングがリバース・エンジニアリングの対象となったプログラムの表現を流用するものであってはならないことはもちろんである。しかし、リバース・エンジニアリングの対象となったプログラムに含まれている著作権法によって保護されない要素のみを利用して別プログラムを開発した場合でも、開発されたプログラムが依拠したプログラムの全体又はその一部において表現が類似してしまうこともあり、しかもこの類似性が、時にはプログラム開発に当たっての技術的制約、あるいはプログラムの効率性や経済性についての配慮、さらには技術の本性としての収斂性に由来することもありうる。そこで、プログラムの著作権侵害訴訟において類似性が問題にされる場合には、類似性の故に安易に侵害を認定するのではなく、プログラムの技術製品及び実用品的特性に鑑み、その開発に当たっての技術的制約、プログラムの経済性・効率性、さらには技術一般の本性から由来するその収斂性などが慎重に考慮されなければならないと考えられる。

この場合、実体法的にみて、他社開発のプログラムに依拠したとしても、そのアイデアにまで遡り、これを利用してプログラムを開発し、その際に独立創作ともいえるほどの知的努力を尽くした結果として当初のプログラムと実質的に類似したものが開発された時には、侵害とはならないと考えられる。ただ実際上訴訟の場における立証の問題として、当初のプログラムと実質的に類似したものが開発された場合に、確かに、まず原告となる当初のプログラムの著作権者が、被告が原告のプログラムにアクセスし、原告のプログラムと実質的に類似するプログラムを作成したと証明すれば一応侵害の推定がなされることになる。しかしその場合でも、次に被告の側で、アイデアまで遡り、且つ先に述べたような諸要因を考慮した上でアイデアを利用して独立創作にも等しい知的努力をなした結果、表現においても実質的に類似するものが開発されたということを反証できる場合には、侵害責任は否定されると考えられる。何故なら、当初のプログラムの表現に対するアクセスがあったとしても、アイデアにまで遡ったことが証明され、且つまたプログラムの開発に当たっての技術的制約あるいはプログラムの経済性・効率性等の要因を考慮した上で独立創作に等しい知的努力がなされた結果として実質的に類似するプログラムが開発されたと証明できる場合には、同一のアイデアから、技術的・経済的な実際上の必然性あるいは技術的収斂性の帰結として、実質的に類似するプログラムが開発されたといえ、そこにはもはや当初のプログラムの表現の流用やただ乗りがあるとはいえないからである。

ただ、やはり実際の訴訟上にこうした反証は往々にして困難なこともありうるが、プログラム開発に伴う諸判断、費用、労力や手続等が文書記録(いわゆる米国で paper trail といわれるもの)として残されている場合には、これについての専門家の鑑定に基づき、裁判官が独立創作といえるに等しい知的努力がなされたか否かを法的評価として判断し、決すべきであろう。また実際、ソフトウェア開発実務の現場では、既存のプログラムのアイデアを参考にしてプログラムを開発する場合に、結果的に実質的に類似するプログラムが作成された時にも、単にアイデアを利用した独立創作であると主張せんがために、既存のプログラムの解析チームと新たなプログラムの開発チームを全く隔離し、解析チームの得た技術的アイデアの授受のみを開発チームに行い、表現の授受を遮断するというクリーン・ルーム方式又はアイソレーション・ブース方式と呼ばれるプログラム開発方法が取られることがある。そして、この手続過程で厳格な分離がなされ、また解析チームの抽出した情報で開発チームにトランスファーされるものにリバース・エンジニアリングの対象となったプログラムの表現が含まれていないかを法律家の立会いのもとに検閲を受け、さらに解析チームにより情報が抽出される過程及び開発チームにトランスファーされた抽出情報が開発チームによってプログラム開発に際しての利用される過程などについて文書記録が残されていれば、上述の意味での独立創作を証明する有効な証拠を残す手続となりえよう。