The Protection of Computer Software under Japanese Law : Status Quo and Suggestions for Future Protection based on Comparative Studies

トラウトマン・ディルク
ドイツ連邦共和国 : 司法修習生

https://doi.org/10.15017/2090

出版情報：法政研究. 64 (3), pp.101-146, 1998-01-21. 九州大学法政学会
バージョン：
権利関係：

NOTES

The Protection of Computer Software under Japanese Law:
Status Quo and Suggestions for Future Protection based on Comparative Studies

Dirk Trautmann

C. Patentability of Software (status quo)

D. Special Problems

I. Interfaces

1) Interfaces and Interoperability

2) The Scope of Protection of Interfaces

a) American Judicial Development

b) Solution for Japanese Law

II. Reverse Engineering

1) Definition

2) Legal Permissibility

a) United States

b) European Union

c) Japan

3) Contractual Protection against Reverse Egineering

III. Microprograms

E. Future Protection

I. The Piracy Problem

II. Examination of Different Systems of Protection

1) Registration Act

2) Patent Law

3) Trade Secret Law

4) Contractual Protection

5) Copyright Law

6) Law Sui Generis

III. Proposal : A Copyright Based Law Sui Generis

F. Conclusion

Appendices

I. BSA Survey on Software Piracy

II. Example of Application Form for Software Registration

III. SOFTIC Statistics on Program Registration (April 30 1996)

Bibliography

A. Introduction

I. General

Computer gained an increasing importance of every part of our life. Computers in this context are not only personal computers (PC) but every type of computer, i.e. including watches, pocket calculators as well as the fastest craigh computer. All these computers follow basically the same principles. It is mainly for practicability reasons and its familiarity if this paper mainly refers to PC's. Computers can only work if there are fitting hardware and software components. The development of hardware as well as software require a high degree of know-how. The necessary high investment made by hardware and software producers would easily be nullified if it were possible for everybody to copy the products without giving a reward to the developer. Thus these products, hardware as well as software deserve protection by the law. While hardware generally enjoys a high degree of protection, as it is technically more complicated to be copied and sufficiently protected by the existing patent law, the situation for software is different.

Demonstrating the system of protection of computer software including its present problems and a future outlook are the tasks of this paper. This paper is only dealing with computer programs, thus leaving aside the problems concerning integrated circuits (semiconductors) and databases[1]. After explaining the factual situation by displaying the technical and industrial background, first the historical development of the copyright protection, which is the main means of protection, will be explained. Afterwards the registration system will be examined, before a brief guide to the status quo of patentability. Thereafter the most problematic points under the present system, such as interfaces, reverse engineering and microprograms will be discussed. Finally the future of soft-

[1] Databases are regulated in the Copyright Act, Art. 2 and 12.

ware protection will be examined.

II. Technical Background

As explained before a computer requires for its use hardware and software components. These components will be explained in turn[2].

1) Hardware

Hardware is the physical equipment of a computer. It mainly consists of three elements : the input/output system, the central processing unit (CPU) and the memory.

The input/output system after receiving information from the outside world, transforms it into (binary) electrical pulses and communicates these pulses to the brain of the computer, the CPU. The input/output system also converts the electrical pulses from the CPU into a usable form and transfers it to the outside world. Typical input/output systems are screen, keyboard, mouse, disc drives and printers.

The CPU, in personal computers, is contained within a single silicon chip called microprocessor. Intel is the most important producer of microprocessors for PC's. The CPU first controls the interaction among various components of the system and the timing of these interactions. Secondly it performs arithmetic and logical operations. Thirdly it accesses the computer's memory.

There are basically two kinds of memory, RAM and ROM. The RAM (Random Access Memory) is the main memory. The RAM holds the program the computer is executing along with some of the program's data. During execution of the program the RAM content changes. Unlike the RAM the content of the ROM (Read Only Memory) is fixed. The programs preserved in ROM are known as firmware because they are stored in the silicon ROM chip and thus have attributes of both hardware and software.

---

[2] For more information see Ron White, How Computers Work (1993).

2) Software

The term software refers to computer programs, which are defined as sets of instructions used in a computer to bring about a certain result. There are two broad types of software : operating system software and application programs.

The operating system is the computer program which controls and coordinates the interaction among the various hardware elements and between the hardware and application programs. Without some form of operating system, the hardware cannot function. The most famous operating system for PCs is MS-DOS (Microsoft Disc Operating System).

Application programs are programs designed to perform particular applications. This includes packaged software as MS-Word or Lotus 1-2-3 as well as individually created software for special tasks. The application software like operating system software must be copied into the main memory before they can be used. Once in the memory, the application interacts with and depends on both the operating system and the hardware.

Software is normally created in a certain procedure. Based on a main purpose, a program structure is created. This program structure is divided into several modules. The modules consist of one or more algorithms and data structures. The last two levels are normally developed by using flow-charts. These flow charts are then translated into a programing language, e.g. Pascal, Basic, Fortran. These programing languages have a syntax close to human languages, which makes them easy to understand. Furthermore the programmer usually adds comments in the program which will not be read by the computer while proceeding the program. The program written in a programing language is called "source code". As the hardware elements of the computer only understand the binary code of electrical pulses, this source code has to be translated into the so called "object code". This translation is carried out by a compiler. In most cases only the object code is given to the user.

III. Industrial Background

The industrial background is important to understand the necessity of an efficient software protection and the scope of such protection. There are two contradicting problems, the problem of illegal copies, requiring strong protection, on the one side and the fact of lock-in, requiring certain limitations, on the other side. A solution has to be found in between these two core problems.

1) Illegal Copies

Programs are usually stored on disc. Technically it is no problem to copy the content of one disc to another disc. As long as this is done for a legitimate reason, e.g. backups, there can be no doubt about its legality. But due to lack of legal understanding as to the consequences copies are made in many further respects. This concerns not only "computer kids" copying games, but is done within all parts of society. Word processing programs are just distributed from one friend to another, companies only buy one copy but use it at several terminals at the same time. In a survey around the world[3] the Business Software Alliance (BSA) estimated the volume of piracy and the consequent loss to its members[4]. Japan is with a level of 80% of piracy one of the leading countries in the world. The estimated loss of BSA members is US $1,961 million.[5] These figures clearly show the necessity of effective legal protection. It also demonstrates that the present law is not sufficient in handling the problem.

2) Lock-in

The question of proper protection is challenged from the other side as well by the so called lock-in problem. It can best be demonstrated at the examples of Intel and Microsoft. Both companies were partners of IBM at the initiation of IBM's development of the PC. Intel distributed the microprocessor and Microsoft the operating system. Because of IBM's great market share the IBM and the IBM-compatible PC became a success. While IBM constructed the case, the most important parts were contributed by Intel and Microsoft. While

---

[3] See Appendix I.
[4] BSA members are : Aldus, Apple, Autodesk, Intergraph, Lotus, Microsoft, Novell, The Santana Cruz Operation and WordPerfect.
[5] IP Asia 30 Nov. 1994, p.2.

other companies could build a case and other hardware parts like IBM, for achieving compatibility it was still necessary for them to use Intel processors and the MS-DOS operating software. Or the other way around : only by using the same application software that was created for IBM PC's could IBM compatible computers be made.

Following this, Intel's processors and Microsoft's operating software set a standard in the PC market. If a user wanted to purchase a new computer while still using her old application software, she had to take care that she was using the same operating software. Therefore Intel and Microsoft gained a monopoly in the market and users were locked-in to their products. As always with monopolies there is a danger of abuse, mainly concerning overpricing. "Thick" protection of Microsoft's operating software will lead to less competition in the same market, thus the probability of abuse increases. Therefore the means of protection are not supposed to be used to perpetuate the lock-in situation, which in turn leads to monopolies.

B. Copyright Protection

I. Historical Development

The following chapter will display the historical background of the present legal situation. Only the development resulting in the amendment of the Copyright Act will be subject of this chapter. Protection under patent law will be dealt with by a separate chapter (part C).

From the beginning of the 1970's there were two positions held within the government. One, represented by the Ministry of International Trade and Industry (MITI), favoured a law sui generis while the other, represented by the Ministry of Education (Monbusho), wanted to achieve the aim of protection by amending the Copyright Act from 1970. Although this dispute was settled with the amendment of the Copyright Act in 1985 examination has not become obsolete. Some of the arguments raised in the discussion may still have some validity for the solution of topical problems under the present legislation.

1) The early stages

In 1972 the Cultural Affairs Agency[6] started setting up a Subcommittee for Computer Problems within the Copyright Council. The decision of setting up the subcommittee within the Copyright Council may have been already a preliminary decision in favour of copyright law. Not unexpectedly in such a framework, the reports of the subcommittee did not touch the question of copyright protection per se but discussed in which respect the Copyright Act was sufficient or had to be amended.[7] The result was mainly that the Copyright Act of 1970 was already capable of protecting computer software in an appropriate way.[8] There were only two additions proposed.

First a distribution right granted to the copyright owner was suggested. Secondly a formal system of deposition or registration was proposed for facilitating the distribution of programs.[9]

About the same time Ministry of International Trade and Industry (MITI) set up a legal committee to study the legal protection of Software. This committee, in May 1972, came to the conclusion that neither copyright law nor patent law was suitable for the protection of software. This committee proposed a registration system for software which at the same time was seen as the appropriate means of protection. MITI's main objective for setting up the committee was facilitating a wider distribution of software.[10] The two different objectives may have been the reason for the different solutions. While the Monbusho approached the problem from the point of protection of software producers, MITI saw the main purpose in promoting industry by making the use of software as easy as possible thus not granting more protection than absolutely necessary.

Although these two basically different opinions arose in the early 70's the dispute did not break out before the mid 1980's.

---

[6] The Cultural Affairs Agency (Bunkacho) is an extra ministerial agency of the Ministry of Education (Monbusho) which administers copyright law and has the Copyright Council as its advisory body.
[7] For details, see Doi, Computer Technology..., p.95, footnote 7.
[8] Doi, supra 7, p.96-98.
[9] Doi, supra 7, p.98.
[10] Doi, supra 7, p.99.

2) Two Different Bills

In December 1983 the Information Industry Committee, instructed by the Industrial Structure Council, an advisory body to MITI, recommended the enactment of law sui generis, called the "Program Rights Law". The MITI gave the following reasons : After consulting about 40 big companies they found out that there is a need for a clear solution of the problem. Patent law was not considered to be suitable as it requires publication and follows the first-to-file principle. Both have a negative effect on the development of software. Publication makes illegal copying even easier. Furthermore it is natural in software production that different programmers come to the same solution independently and both are deserving protection by the law. Copyright law was also not seen as being an appropriate measure as its main subject is the protection of works of art and culture. Software on the other hand is mainly commercially based. This commercial argument was the decisive for the MITI. The "Program Rights Bill" had not only the aim of protecting software producers but was also supposed to promote and facilitate the development, distribution and use of computer programs, by avoiding multiple investments (time and money) into the same development.

Central means of protection was the "using right", granted upon creation of the program.[11] An obligatory registration system should be introduced. A summary description of the registered program was to be published.[12] The right was granted for 15 years[13] (much shorter than the 50 years under copyright protection). In specified cases by using an arbitration system the holder of the right could be forced to license his program.[14]

This bill was criticized domestically and abroad.[15] Following its former point of view the Bunkachou argued a separate law was unnecessary as a computer program is a copyrightable work and existing uncertainties could be

[11] Art. 3 par. 1, Art. 4 of the Bill.
[12] Art. 6.
[13] Art. 5.
[14] Art. 8.
[15] For more details see Rahn, Sonderschutzgesetz fü Computerprogramme in Japan ?, GRUR Int. 1984, 217, 219.

clarified by amendment of the copyright law. The argument of economic importance was rejected. Also for other works protected under the copyright law the commercial value is important, e.g. publishers. Computer programs were used in all areas of life so that it also plays an influential role in the cultural development of society. Furthermore almost all other developed countries had introduced copyright protection for software.

The criticism from abroad came mainly from the United States. The United States feared that the "Program Rights Law" would favour Japanese software industry by putting the American and European software industry into a disadvantageous position. The fears related to the short term of protection and forcing parties to enter into license agreements if no agreement could be reached by normal negotiations. The United States warned Japan that they would consider counter measures in case the "Program Rights Bill" became effective.[16] On the other hand the U.S. did not object to protection under the copyright law. Although the U.S. criticism was rejected by the MITI, the American point of view becomes understandable if one considers the MITI's aim of promoting domestic industry.

During the ongoing discussion three judicial decisions concerning video[17] games were rendered in which the courts established that copyright protection existed for computer software. This may be an additional reason why an amendment of the Copyright Act was finally adopted.

II. The 1985 Amendment of the Copyright Act

Although the 1985 amendment of the Copyright Act is often regarded as a victory of Monbusho[18], the outcome is actually more of a compromise between the two positions.[19] Except that copyright was adopted as the general means of

[16] Rahn, p.219.
[17] "Taitou Space Invaders", Tokyo District Court, Mutaishuu Nr. 14-3, 796 ; "Super Invader", Yokohama District Court, Hanrei Jihou Nr. 1081, 125 ; "Strategy X", Osaka District Court, Mutaishuu No. 16-1, 26 ; all displayed in Doi, supra 7.
[18] Sommer, Die Schutzfäigkeit von Computerprogrammen nach japanischem Recht GRUR Int. 1994, 383, 387.
[19] As here e.g. Pilny, Die Registrierung von Computer Software in Japan, GRUR Int. 1988, 26 (27).

protection, there are several points in which the MITI's point of view was in fact adopted. For this purpose several provisions which contain special regulations for program works were introduced into the Copyright Act.

Art. 2 (1) (x) defines "program" as an expression of combined instructions given to a computer so as to make it function and obtain a certain result. Art. 10 (ix) adds "program works" to the list of works protected by the Act. Art. 10 (3) excludes "any programming language, rule or algorithm used for making such works." This is followed by a definition of the terms "programming language", "rule" and "algorithm". This exclusion follows a fundamental principle of copyright law, whereby an expression can be protected, but neither the underlying idea nor the necessary means or tools for its creation.

Although Art. 20(1) generally grants a right of preserving the integrity of a work, there is an exception concerning program works in paragraph (2). The right of preserving the integrity is not applied to a "modification which is necessary for enabling to use in a particular computer a program work which is otherwise unusable in that computer, or to make more effective the use of a program work in a computer".

Art. 47bis gives a special right of reproduction and adaptions to the owner of a copy of the program work "if and to the extent deemed necessary for the purpose of exploiting that work in a computer by himself, provided that the provision of Art. 113 (2) does not apply to the use of such copies in connection with such exploitation." Nevertheless this right ends when the owner of copies has ceased to have ownership of any legal copies for reasons other than those of destruction.

Art. 20 and 47bis clearly fulfill the MITI's aim of facilitating the use and development of existing software by applying some less restrictive rules to software. This was also intended for the registration system introduced in Art. 76bis and the Registration Act based on Art. 78, Art. 78bis. As the next chapter is dealing with the Registration system in detail there is no need for a further description at this stage.

On the other hand protection is exceeded by Art. 113 (2) : "An act of using in a computer, in the conduct of business, copies made by an act infringing a copyright in a program work (including copies made by the owner of such copies in accordance with the provision Art. 47bis (1) as well as copies of a program work imported ... and copies made by the owner of such imported copies in accordance with the provision of Art. 47bis (1)) shall be considered to constitute an infringement on that copyright, so long as a person using such copies is aware of such infringement at the time when he has acquired an authority to use these copies."

Concerning the authorship of a work made by an employee in the course of his duties, program works are an exception to the general rule of Art. 15 insofar that it is automatically attributed to the legal person or employer on whose initiative the program was created automatically unless otherwise stipulated in a contract, work regulation or the like in force at the time of creation (Art. 15 (2)). The reason for the exception from the requirement of making the program public under the name of the employer is that programs are often not made public at all or if made public not under a certain name.[20]

Due to the general principles of copyright law, a program work must fulfill the following requirements to be protected under the Copyright Act :

(a) Thoughts or sentiments must be expressed in the work
(b) These thoughts or sentiments must be manifested in an expressed form (no special medium is required).
(c) The expression must be creative or original.
(d) The product must fall within the literary, artistic, musical or - as software regularly does - scientific domain.

III. Court Decisions after the Amendment

There are only a few court decisions concerning copyright protection in Japan. Two of them will be discussed here, while another will be explained at a later stage (D. III. Microprogram).

[20] Negishi, p.413.

1) Microsoft v. Shuuwa

The first significant case decided after the amendment was Microsoft Corp. v. Shuuwa System Trading K.K.[21] The plaintiff's operating system was written in assembly language, converted into object code and stored in the ROM. The defendant disassembled the object code, attached labels and comments and finally published the work in book form. The Tokyo District Court found an obvious infringement of the plaintiff's copyright. By ascertaining a copyright protection for the plaintiff's operating system, this ended the discussion as to whether operating software is protectable as a program work or not[22].

2) System Science v. Toyo Sokuki

The plaintiff in System Science K.K. v. Toyo Sokuki K.K.[23] developed four programs for biochemical measurement. The defendant copied three of these programs and installed them into ROM. This was a clear copyright infringement. The interesting point was concerning the fourth program, which the plaintiff alleged was adapted by the defendant to create a new program. The High Court stated : "In order to be able to decide whether a certain program infringes the copyright in a program work, it is of course necessary that there is a combination of instructions in the program work that can be found to be creative, and that the combination of instructions of the later created program is similar to the part of the program which can be found creative...[24] After adopting this rule to the present case the Court found that there was no infringement as the similarities of the programs existed in expression limited by hardware constraints, commonplace or standard expression and expression dictated by common sense. It also confirmed that the processing flow of a program is not protected since "algorithms" are excluded by Art. 10 (3) Copyright Act.

IV. The Registration System

The registration system is one of the peculiarities of Japanese software protection under copyright law. Based on the Articles 76bis, 78 and 78bis of the

[21] Tokyo District Court, 1219 Hanji 48 (Jan. 30, 1987).
[22] Band/Katoh, Interfaces on Trial, 289 ; Pilny, Schnittstellen ..., GRUR Int. 1990, 436ff.
[23] Tokyo High Court, 1322 Hanji 138 (June 20, 1989).
[24] Id. at 140.

Copyright Act, it is further regulated by the Law on Exceptional Provisions for the Registration of Program Works ("Registration Act").

1) Legal Nature and Reasons

Art. 76bis of the Copyright Act

(1) The author of a program work may have the date of creation of his program work registered, provided that a period of six months has not passed since the creation of that work.

(2) Program works as to which the date of creation is registered in accordance with the preceding paragraph shall be presumed to have been created on the date registered.

First the legal nature of the registration has to be explained. As for all other works protected by copyright law, copyright is granted upon creation, thereby existing regardless whether the program is registered or not. A right granted upon registration would constitute a violation of Art. 5(2) of the Berne Convention to which Japan is a party. The exclusion of any formality is also guaranteed in Art. 17 (2) Copyright Act. As registration is not necessary for obtaining copyright the question is what advantages does registration have.[25]

The first advantage of registration is a facilitation of prooving of the existing right in infringement cases. Therefore Art. 76bis (2), which contains an assumption that the program was created on the day it is registered, plays a central role. Secondly the identification of the program is possible without any problems. The question of identification arises when the author wishes to transfer any rights on the program. By using the registration number identification becomes clear to everyone. Furthermore the author expresses the will to secure her rights and that she is the owner of the rights.

So far the incentives for an author to have the program work registered. Furtheron there were structural and public reasons for implementing the registration system. As shown above the MITI wanted to implement a law sui

[25] See SOFTIC, Guide to Program Registration, p.1 ; Pilny, Die Registrierung..., GRUR Int. 1988, 26, 28.

generis concerning computer software for facilitating the use and development of software. By insisting on a registration system MITI wanted to achieve this aim in the following two ways. First from the number of applications the public could estimate a company's economical and inventory strength. Secondly information is provided for the professional world. Therefore other developers do not have to make the same research and development as the first developer as they can get information about what was developed. This, of course, does not mean that they have access to the registered programs but they can approach the author after obtaining the specification from the registration gazette.

To which extent the intended goals were achieved will be examined later (III) after an explanation of the registration procedure.

2) Procedure

a) The Registration Institution

Art. 78 Copyright Act provides that the Commissioner of the Agency of Cultural Affairs is keeping the register, but Art. 5 (1), 7 Registration Act gives the right of partial or complete designation of the registration business to a legal entity. This was done December, 17 1986 by delegating the registration business to the Software Information Center (SOFTIC, sofuto jouhou senta). Since April 1 1987 registration is only possible at SOFTIC. SOFTIC is a foundation (zaidan houjin) in compliance with Art. 33 ff. Civil Code. It is mainly financed by membership fees and not by registration fees.

SOFTIC has three departments : registration department, administration department and a research department. The last department indicates that SOFTIC is not only keeping the software register but also provides academic research facilities. For example every two years an international symposium concerning legal problems of computer software is organized by the center.

b) The Registration Procedure

The author of a program work has to submit the following materials for registration.[26]

[26] For details see SOFTIC, Guide to Program Registration.

(a) Application form containing the title, type and date of creation of the program as well as address and name of the applicant.[27]

(b) Specification of the program including description. This description must not have more than 200-400 characters. Furthermore the applicant has to classify his program by selecting one of the 75 categories provided by SOFTIC.

(c) Copy of the program on microfiche.

(d) Proof that the required registration fee was paid.

Before registration the application is only examined concerning formalities. For example the description of the program may only be rejected if it is not understandable or illogical.

If the applicant fulfilled all the formalities the program will be registered. Afterwards the registration number, registration date, name and address of the applicant as well as the description and the classification of the program are published in a monthly gazette as provided by Art. 4 Registration Act. Any person may demand the delivery of copies of entries in the program register after paying a fee (Art. 2 (2), (3) Registration Act). These copies only include the published facts, not the copy of the registered program itself.

3) Acceptance by Software Producers

As mentioned above registration is not necessary for protection by copyright. On the other hand the aims targeted by the MITI can only be achieved if as many software producers as possible have their programs registered. From the beginning SOFTIC expected 100 applications per month.[28] As to be seen from the statistics dating from April 30, 1996[29] for the fiscal years 1987 to 1995 the number of registration is constantly between 530 and 602, with an even lower number (499) for 1988. Almost no programs (so far 35 in 9 years) were registered by companies from abroad. About 90% of the registered programs have the date of creation registered, while the registration of the date of first publication, of the true name and of the copyright (in case of copyright transfer,

[27] Sample of application form see Appendix II.
[28] This was the amount calculated in the budget. See Pilny, Die Registrierung von Computer software in Japan, GRUR Int. 1988, p.26 (32).
[29] Appendix III.

etc) play a minor role. This means that as intended Art. 76bis (2) with its assumption for the date of creation is the main incentive for registration.

While this incentive is important the main question remains unanswered. Why is the interest in registration much lower than expected ? For the answer a look at criticism by industry and legal scholars is helpful.[30]

a) Criticism by the Industry

One of the aims of the registration system is to provide information about technical development and facilitation of the development of software. The published information about the registered programs (the short description) is in most cases insufficient due to the limitations of 400 characters. Furthermore big system-developers concentrate more in the production and selling of hardware, while software users (companies) often have very specific and unique tasks, so that they cannot expect to find an existing program for their tasks. Another factor is that in no other technical field are there as many publications about technical development as in the field of computer soft- and hardware.

Although the assumption of Art. 76bis (2) seems to be a strong incentive a lot of companies do not consider registration as being necessary to prove the date of creation. Especially bigger software developers have detailed "in-house documentation regulations", whereby they can easily prove the different stages of development.

There may also be some fear whether copies submitted with an application are really secure at SOFTIC. Accordingly it is probable that the most inventive programs are not registered.

Last but not least industry may not be too happy that the Bunkacho is dealing with software matters instead of the more industry related MITI.

[30] See Pilny, supra 28, 33.

b) Criticism by Scholars

Scholars have criticized that the determination made by Art. 76bis cannot be made, as there is ongoing development even after selling the program.[31] The practice of just having a few modules of the whole program registered the determination of Art. 76bis was also challenged. The question is whether the registered modules are representing the whole program or whether they can constitute a separate ("central") program work itself.[32]

A positive remark was made concerning the fact that the registration is a certification of who is the owner of the right. This is of importance especially for licensees, who normally cannot be sure whether the licensor is the owner.[33] This argument has to be doubted as only formalities are checked in the application process by the SOFTIC prior to registration. Furthermore registration may not be changed although the rights in the program had been transferred.

It may be concluded that the registration system has brought the protection of software closer to that of patent protection or protection by law sui generis. However as this has not been done consistently, doubts remain as to the benefit of the regulation.

C. The Patentability of Software (status quo)

As copyright only protects expression and in practice it is often difficult to determine what is an expression and what is an idea, copyright may not be the only way to protect software. As an idea may be a technical invention in this chapter it will be examined to what extent software related inventions are patentable in Japan.

The patent law of Japan does not distinguish computer-software-related inventions from other inventions, and thus the general principles of patentability

[31] Pilny, supra 28, p.35.
[32] Id.
[33] Id.

apply equally to any kind of inventions. However, inventions of this technical field often consist of a mix of technical and non-technical features, tending to cause difficulties in judging whether claimed inventions fall within the statutory intention.[34] Therefore the Patent Office released examination guidelines. First initiated in 1975 the guidelines have been revised in 1982, 1985, 1989 and most recently in 1993. These guidelines will be explained in turn.

As the Patent Law Section 2 provides that "invention" means "a highly advanced creation of technical idea utilizing natural laws" this is further explained in the guidelines. If natural laws are utilized in information processing by software, a claimed invention is deemed as utilizing natural laws.[35] Even if natural laws are not utilized in information processing by software, a claimed invention is deemed as utilizing natural laws provided that it utilizes hardware resources. This does not mean that any software utilizing hardware resources is patentable, it must still fulfill the requirement of a "technical idea".[36] It is important to mention that an invention claimed as a "computer program" or "software" does not constitute the statutory invention irrespective of the contents of the program since a computer program as such is not regarded as a patentable subject matter. In the case of software the guidelines are referring to copyright protection.

Further patentable software-related inventions are[37] :

- Operating Systems, which control the action of the computer itself
- Interfaces between machines and human beings
- methods for controlling multi-programming
- procedures for information processing based on physical or technical specifications of the object, e.g. processing of pictures by computer
- although "linguistic processing" is not mentioned a "kanji-kana-conversion method" is given as patentable example

Although most of the discussions concerning the protection of software are

[34] Yoshiaki Aita, Patentability of Computer-Software-Related Inventions in Japan, 4th Int. SOFTIC Symposium, 303.
[35] Id. at 299.
[36] Sommer, supra 18, 390.
[37] See Sommer, supra 18, 390.

in the field of copyright, patent protection plays an important role. For instance, the number of patent applications under the International Patent Classification G06F, digital computers was 16,000 in 1986, 19,000 in 1988 and 21,000 in 1990.[38]

For an evaluation of the reasonableness of patents for software see the chapter about future protection.

D. Special Problems

I. Interfaces

The problem of protection of interfaces as well as the question of reverse engineering are based on the question of interoperability. First the terms interface and interoperability deserve a further explanation. Afterwards it will be examined to what extent interfaces are protected by copyright. This is one of the key points to determine the scope of copyright protection.

1) Interfaces and Interoperability

As explained in the introduction a computer system consists of separate hardware and software elements. These elements have to be connected in a certain way, so that they can interact with each other without requiring other connected products to alter their mode of operation. There must be interoperability or connectability. The functional characteristics of any element's interoperability with the rest of the computer fall into three categories : the controls to the element, permissible inputs into the element and permissible outputs from the element. These functional characteristics are the element's interfaces. If the output from WORD is not permissible input into MS-DOS, they have incompatible interfaces, that means they cannot interact properly. There are hardware/hardware interfaces (e.g. between CPU and the main memory), software/hardware interfaces (e.g. MS-DOS has to work with the PC hardware), software/software interfaces (e.g. WINDOWS working under MS-DOS, WORDPERFECT working under WINDOWS). The category of hardware/hardware interfaces is of no interest here, as this paper is limited to software protection.

[38] 4 th Int. SOFTIC Symposium, p.30.

Regardless of whether the interface is concrete (a specific instruction) or abstract (the syntax and semantics of permissible inputs), it can be described in great technical detail in an interface specification. These interface specification are needed to develop interoperable products.

2) The Scope of Protection of Interfaces

The question to what extent interfaces are protected by copyright law is of vital importance. On the one hand side are the producers who develop interfaces. These producers invest a lot of money in the research and development of such interfaces. They argue that copyright protection just guarantees the reward for their investment. There would be no incentive for the production of new interfaces if this reward could not be obtained.

On the other hand side are consumers and producers of interoperating, compatible software. They argue that it is an overprotection which would lead to monopolization. This is where the lock-in effect (see A. III) plays its most important role. If for example all interface specifications of WINDOWS were protected, only Microsoft or Microsoft's licensees could develop and sell compatible application software. They argue that overprotection of interfaces would restrict competition to an unacceptable extent.

The discussed decision System Science v. Toyo Sokuki touches the crucial point of the legal protection of interfaces, but as it is not dealing with interfaces, there are no decisions by Japanese courts concerning interfaces yet. Therefore the discussion about Japanese protection is often unclear. Some commentators simply judge from the existence of the exceptions of program languages, algorithms and especially rules from copyright protection (Art. 10 Copyright Act) that interfaces are never protected.[39]

On the other side of the spectrum it is argued that all interfaces are generally protected. This position is gained from a discussion about so called "user interfaces".[40]

[39] Band/Katoh, supra 22, p.288 ; Ozaki, Copyright Protection of software : The Japanese View, 1990 Computer L. Rep. 950, 959 ; Nobuhiro Nakayama, Legal Protection of Software (1988) 42.
[40] Hirakawa, Nakano, Copyright Protection of Computer Interfaces in Japan ; 1990 EIPR 46, 48.

Thus first the point of discussion has to made clear. "User interfaces" are the connections between the computer system and the human being, the user. Examples are printer, mouse, keyboard. The discussion about user interfaces is mostly concerning the "touch and feel" of screen compositions. These are primarily protected as cinematographic works under Art. 10 (1) (vii)[41]. Although the program creating a certain screen can be protected as a program work as well, this is not the crucial point. The decisive thing in this discussion is whether the appearance is the same. The same appearance can be generated by totally different programs. Although the problem of protection of "touch and feel" is of great interest, the discussion has to be led on the different level of cinematographic works and is therefore omitted here. Consequently this paper is only concerning the protection of hardware/software and software/software interfaces.

a) American Judicial Development

Due to the lack of Japanese court decisions a look at the judicial development in the United States may be helpful.

Whelan v. Jaslow[42]

In the often cited Whelan decision the plaintiff was hired by the defendant to write a program to run a dental laboratory. While the plaintiff was granted the copyright, the defendant was supposed to market the program to other dental laboratories. Finding out that the plaintiff's program was based on a rarely used system, the defendant wrote a new program in BASIC. The plaintiff claimed copyright infringement.

The defendant's program was not a translation of the plaintiff's work, but for example file structures, screen outputs and some subroutines were used which performed similarly. The Court examined the expression/idea dichot-

[41] Id. at 51 ; although they consider also a protection as program work, as an annex the Tokyo District Court in K.K. Namco v. K.K. Gijutsu Hyoronsha (EIPR 1994, D-202) concerning the computer games Pacman and Ms. Chomp just considered protection as cinematographic work.

[42] Whelan Associates, Inc. V. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3rd Cir. 1986), cert. denied, 479 U.S. 1031 (1987).

omy and came to the conclusion, that the idea of the program was "the efficient management of a dental laboratory.... Because that idea could be accomplished in a number of different ways with a number of different structures, the structure of the [Whelan] program is part of the program's expression, not its idea."[43]

The court therefore developed the "one idea principle", which as its consequence had a wide area of expression, including all interface specifications. This led to a very "thick" protection of software.[44]

Computer Associates v. Altai

This principle was revised in Computer Associates v. Altai[45]. The court acknowledged that "each subroutine is itself a program, and thus, may be said to have its own "idea",..."[46]. The Court used a three step similarity test[47]. The first step is abstraction. "This process begins with the code and ends with an articulation of the programs ultimate function."[48] The second step is filtration. This means "examining the structural elements at each level of abstraction to determine whether their inclusion at that level was "idea" or was dictated by considerations of efficiency, so as to be necessarily incidental to that idea ; required by factors external to the program itself ; or taken from the public domain and hence is nonprotectable expression."[49] The Court later pointed out which circumstances are limiting the programer's freedom of choice so much, that they are not protected :

"..(1) mechanical specifications of the computer on which a particular program is intended to run ;
(2) compatibility requirements of other programs with which the a program is designed to operate in conjunction

---

[43] Id. at 1234.
[44] Band/Katoh, supra 22, p.95.
[45] Computer Associates Int'l, Inc. v. Altai, Inc. 775 F. Supp 544 (E.D.N.Y. 1991), affd in part, rev. in part, 982 F.2d 693 (2nd Cir. 1992).
[46] Id. At 705.
[47] See figure, Appendix IV.
[48] Id. at 707.
[49] Id.

(3) computer manufacturers'design standards
(4) demands of the industry being served
(5) widely accepted programming practices within the computer industry."[50]

From point 1 and 2 it follows that the Court did not consider the interface specifications as protected expression.

The final step is a comparison of the protected parts of the program with the allegedly infringing program. If there are similarities the program is infringing the plaintiff's copyright. In the present case the Court found all similarities to be factors external to the program itself, thus not be protected by copyright.

This case in its decision and the reasoning is similar to the aforementioned Japanese case System Science v. Toyo Sokuki. However the American decision hands down a clearer guideline for measuring software protection.

Atari v. Nintendo

The decision rendered in Atari v. Nintendo[51] was based on the guidelines given in the Computer Associates decision. The case concerned a Nintendo program which prevented an unauthorized game cartridge from being played on a Nintendo game console ; if the software in the game cartridge did not generate the proper data stream, it could not open the "lock" in the console. Atari had copied elements the Nintendo "key" program so that Atari game cartridges could run on the Nintendo console. In the second step of analysis, the filtration, the Court found that Nintendo used in its "program creative organization and sequencing unnecessary to the lock and key function."[52] This implies that the key-lock function itself was not protected, but the creative expression used by Nintendo. Thus, by just copying the Nintendo program Atari infringed Nintendo's copyright.

Sega v. Accolade

The facts of Sega v. Accolade[53] have some similarities to Atari v.

[50] Id. at 709, 710.
[51] Atari Games v. Nintendo of America, Inc. 975 F.2d 832 (Fed. Cir. 1992).
[52] Id. at 840.
[53] Sega Enterprises, Ltd v. Accolade, Inc., 977 F.2d 1510, 1524 (9th Cir. 1992).

Nintendo as Accolade like Atari used the key created by Sega for Sega game consoles. The significant difference in this case was that there was only one key fitting to the lock : "20 bytes of initialization code plus the letters S-E-G-A."[54] Thus the court by using the method developed by the Computer Associates Court found that there was no freedom of choice for the programmer, thus the use of the key could not infringe any copyright.

b) Solution for Japanese Law

To find the proper solution concerning the protection of interfaces one has to distinguish the interface approach used by the application from the interface specification. The interface specification (in its abstract form) just describes the idea behind the interface and can thus not be protected by copyright. The interface on the other hand is the actual use of the specification. Technically the basis on which the application is supposed to work, hardware or operating software, has to be looked at. Theoretically the lock has to be inspected and the appropriate key has to be construed by the interoperable developer himself. This means where there are several possible keys the interoperable developer is prevented from copying the original developer's interface expression.

This seems also the conclusion to be drawn from Sega v. Accolade and Atari v. Nintendo.

However this solution is contested by some scholars. They argue that since already the development of the interface specification requires a great investment of know-how and money, a protection of this investment is necessary. This financial argument cannot withstand a test of the standards of copyright law. Even more, the purpose of the Copyright Act is not to grant monopolies but as Art. 1 states : "... to secure the protection of the rights of the authors, etc. having regard to a just and fair exploitation of these cultural products, and thereby to contribute to the development of culture." The word "culture" in this context also comprises computer programs as just the 1985 amendment explicitly mentioned program works and in this context Art. 1 was

[54] Id. at 1524.

not changed. For a development in the field of software compatibility is a decisive factor. Furthermore the truth of the financial argument has not yet been proved. One has to consider that a product becomes more interesting to the user the wider the use is. Thus if there are more games for Nintendo consoles, people more likely buy the console. This means there may be less profits for Nintendo from selling games but on the other they might sell more consoles. For these reasons the solution of not protecting the (abstract) interface specification seems to be appropriate.

How to get the abstract form of the specification is most times a question of reverse engineering. This question will be discussed in due course.

II. Reverse Engineering

1) Definition

Reverse Engineering consists of two parts : the reverse analysis and the use of this analysis in the development of another program, forward programming. In using reverse engineering a programmer tries to understand another program. As explained before, usually only the object code of a program is available. This object code, consisting of "0" and "1" cannot be understood by a human being. Therefore several techniques of reverse engineering have been developed. One technique is blackbox reverse engineering, i.e. the programmer performs certain test runs with the program and looks at the externally visible characteristics without looking into the program itself. A step further is the use of memory dumps, i.e. the programmer has the memory displayed at a certain stage to understand the program. The last means which most of the discussion on reverse engineering is about, is decompilation or disassembling of the object code into source code, i.e. transferring the object code into an understandable form. This is not as easy as it sounds. First the level of the decompiled code is still a low level code, closer to assembly language than to higher programming language. Secondly in the compilation process all the comments made by the original programmer are omitted. They cannot be regained in the decompilation process. Therefore reverse engineering, especially disassembling is normally just used as a last means.

For interoperable developers reverse engineering is of crucial importance.

In a lot of cases they are not able to obtain the interface specifications necessary for the development of a compatible program, since the developer of the original program wants to make profits by selling his own applications without severe competition. To what extent reverse engineering is permissible is the question of this section.

2) Legal Permissibility

Until now the legal situation concerning reverse engineering is widely unclear. Only the aforementioned decision Microsoft v. Shuuwa discusses part of the problem ; there is no legislation on reverse engineering yet. Therefore first an analysis of the situation in the U.S. and in the European Union will be made.

a) United States

In the United States the problem of reverse engineering is discussed under the fair use doctrine. The two most important decisions in this field are the already under heading of interfaces discussed cases Atari v. Nintendo and Sega v. Accolade. But before those judgements the courts had to decide whether already the act of loading the program into the memory for doing reverse engineering constitutes a copyright infringement. In Vault v. Quaid[55] the plaintiff argued that already the copying into the memory constitutes copyright infringement as Section 117 permits making a copy when "it is created as an essential step in the utilization of the program." The plaintiff argued that this has to be interpreted as the purpose intended by the vendor. The Court did see no reason for such a limited interpretation of Section 117 and thus decided that copying the program into the memory did not infringe the copyright.[56] Consequently all forms of reverse engineering except disassembling were permitted by this decision.[57]

In Sega v. Accolade[58] the Court explicitly said that disassembling is far

[55] Vault Corp. v. Quaid Software Ltd., 847 F.2d. 255 (5th Cir. 1988).
[56] Id. at 261.
[57] Band/Katoh, supra 22, 171.
[58] Sega Enterprise Ltd. v. Accolade, Inc. 977 F.2d 1510.

beyond the exception of Section 117 and thus can only be justified by the fair use doctrine of Section 107. This provides four factors by which the action has to be measured :

(1) the purpose and character of the use, including whether such use is of commercial nature or is for nonprofit educational purposes
(2) the nature of the copyrighted work
(3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole ; and
(4) the effect of the use upon the potential market for or value of the copyrighted work.

As to the first factor the Court found that the fact that Accolade used the result commercially is presumably unfair, but can be justified by the characteristics of this use. The Court said that Accolade did not sought to avoid performing its own creative work and that no other method was available to Accolade. Thus disassembling was made for a "legitimate, essentially non-exploitative purpose.."[59] Furthermore it was only done for discovering unprotected parts of Sega's program.[60] Another aspect was the development of new creative works which was public benefit.[61]

Concerning the second factor the Court made clear that Sega's program was not "unpublished" in the sense of the Copyright Act since it was sold to the public, and therefore did not deserve a higher degree of protection.[62] The third factor was dispensed with the argument that the ultimate use (opposed to direct use) was very limited and thus of little weight.[63]

As to the fourth factor, the Court found that Accolade did not plan to compete with a certain Sega game but just wanted to extend the market for games played on Sega consoles.[64] Furthermore the Court said, that otherwise

[59] Id at 1522.
[60] Id.
[61] Id at 1523.
[62] Id at 1526.
[63] Id at 1526-1527.
[64] Id at 1523.

Sega's monopolization would run counter to the statutory purpose.[65] Therefore Accolade's disassembling was decided to be fair use and thus permitted.

In Atari v. Nintendo[66] the Court also considered disassembling by Atari as fair use. The Court reasoned that "when the nature of a work requires intermediate copying to understand the ideas and processes in a copyrighted work, that nature supports a fair use..."[67] As it is impossible to understand the object code without disassembling this method is necessary to facilitate understanding the idea or process expressed,.."[68] On the other hand "the fair use reproductions must not extend exceed what is necessary to understand the unprotected elements of the work."[69] As Atari went beyond this line by copying part of the program this was not to be found fair use.

These two decisions give a clear guidance to what extent reverse engineering is permissible.

b) European Union

In the European Union the question of reverse engineering is regulated in Art. 6 of the EU directive on the Legal Protection of Computer Programs. Art 6 permits decompilation "if it is indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs, provided the following conditions are met :"

The decompiling programmer must have a right to use a copy of the program, the information necessary for interoperability has not been already available to her and decompilation is limited to the part necessary to achieve interoperability. Paragraph 2 further limits the right of decompilation for mere interoperability purposes.

---

[65] Id.
[66] Supra 51.
[67] Id at 843.
[68] Id at 844.
[69] Id at 843.

On the one hand Art. 6 permits the decompilation for developing as well as attaching programs. On the other hand the wording leads to the assumption that only software/software interoperability can be achieved, but not software/hardware compatibility (like in the cases of Atari and Accolade). However it is widely agreed upon the intention of the legislator not to exclude hardware/software interoperability.[70] The permissibility of reverse engineering is thus similar to the United States although the technique used is different.[71]

c) Japan

Since in the U.S. by court decisions and in Europe by the EU directive some standard was set for the permissibility of reverse engineering, the question is, whether Japan has such a standard which can be determined by interpretation of the Copyright Act. First the black box reverse engineering will be highlighted before disassembling will be discussed.

The problem of black box reverse engineering is whether the copying and loading into the computer for this purpose is permitted or not. The incidental copying associated with the use of the program is not a reproduction within the meaning of the Copyright Act[72], or even if it is a reproduction it is clearly permitted under Art. 47bis (1). Therefore black box reverse engineering is permitted under Japanese Copyright Law.

Another question is the permissibility of reverse engineering in the form of decompilation. Japan does not have a fair use doctrine like Section 107 of the American Copyright Act, but Art. 1 Copyright Act defines the purpose of the copyright law as "to give protection to the rights of author while giving

[70] Michael Lehmann, The European Directive on the Protection of Computer Programs, in A Handbook of European Software Law 165, 178 (1993).

[71] For the question concerning the international effect of this regulation see Thomas Heymann, The International Effect of the EU Restrictions On Reverse Engineering, The International Computer Lawyer 1994, 15 : The EU regulations will apply to all software sold in the EU, regardless to what extent the country where the reverse engineering was done allows such a practice.

[72] Report by the Second Subcommittee of the Copyright Council of the Cultural Agency 22 (1973) ; Interim Report by the Sixth Subcommittee of the Copyright Council of the Cultural Agency 50 (1984).

consideration to fair utilization of these cultural products, and thereby contribute to development of culture". This means fair use is accepted by the Copyright Act. It is not determined how to measure fair use. Art. 10 (3) excludes program languages, algorithms and rules from protection. It follows that these are considered public domain. As often only the object code is provided, this public domain can only be discovered by reverse analysis. If reverse engineering was not allowed, there would be a de facto protection of rules and algorithms. This is contrary to the intention of Art. 10(3).

As there are no factors given for the determination of fair use, this can be adjusted to the specific problem. In the case of reverse engineering it is appropriate to distinguish the two levels of reverse analysis and forward programming. The reverse analysis should be completely free. This is not a more liberal approach than the protection granted for literary works. The difference is that it is no problem analyzing a book, while a program written in object code cannot be understood. The question then is only what is allowed to be done with the results ? At this stage it has to be made clear that no infringement of copyright takes place. The unprotected parts of a program may be used for the development of new programs or for publication but the protected parts may not be used. Thus in forward programming the developer is advised to use a clean room method. This means the team developing the new product do not have any information about the reverse engineered software except what is given to them from the reverse engineers. The reverse engineers thus have to make sure to give only unprotected material into the clean room.

In publishing the whole Microsoft Program the defendant in Microsoft v. Shuuwa[73] thus crossed this borderline. This is why the court found copyright infringement. Although from this decision some scholars draw the conclusion of unlawfulness of reverse engineering in Japan, the court did not consider reverse engineering per se but all the acts together as an infringement of

[73] Supra 21.

copyright.[74]

It should be mentioned that the Cultural Agency tried to reach a legislative solution. For this reason the Agency asked the Research and Cooperation Committee on Issues Related to Computer Software, chaired by Prof. Zentaro Kitagawa, to work out a recommendation on reverse engineering.[75] Mainly due to pressure from the United States which wanted Japan to ban reverse engineering, the Committee changed its opinion from favouring reverse engineering for extracting "ideas" behind a program to differing opinions and accordingly did not reach a consensus.[76] Thus the Committee did not make any recommendations and the situation has still not been clarified.

3) Contractual Protection against Reverse Engineering

Software companies try to protect themselves from having their products reverse engineered on a contractual basis. Often so called "shrink-wrap contracts" are used. These contracts contain a clause stating that reverse engineering is prohibited. The question is whether these contractual prohibitions are valid. For the answer one has to look at the way those contracts are concluded. The license agreement is written on the envelope containing the discs. As soon as the customer opens the envelope she accepts the terms of the agreement. There is no way for her to negotiate the terms of this agreement, it is just a "take-it-or-leave-it contract" in which the software developer has all the bargaining power. As in most other developed countries Japan has measures of restricting the effect of such standard contracts.[77] By applying these means certain contra legem effects of a shrink-wrap contract will not be permitted and the contract will be void in this respect. Therefore the contractual prohibition of reverse engineering in shrink-wrap licenses do not have the intended legal effect. Nevertheless, the situation may be different in cases where the contract was individually negotiated.

---

[74] Sugiyama, Reverse Engineering, World Computer Law Congress 1991, 2 ; Nakayama, The Legal Protection of Software (1988), note 99, at 131-132.
[75] World Intellectual Property Report 1993, 263.
[76] World Intellectual Property Report 1994, 199 ; 1993, 331.
[77] See Hisashi Tanikawa, Standard Form Contract in : The Japanese Legal System, 132.

III. Micro Programs

Another unsettled problem of software protection is demonstrated by ICM Corp. v. Met's, Inc.[78] The plaintiff owned a program called "EO System" which included a series of files. The defendant developed a program with similar functions as "EO System". The plaintiff claimed the defendant infringed the copyright on the EO System's Install Batch Files file (IBF-file). The IBF file consisted of nine elements : (1) An ID line ; (2) a title line ; (3) a device line ; (4) a pause mark ; (5) a command line ; (6) another pause mark ; (7) a message to users to insert discs ; (8) an installation message line ; (9) an end mark.

The Tokyo District Court ruled that the IBF-file is not protected by copyright law as its limited expression does not fulfill the creativity requirement.[79] The Tokyo High Court went even further and negated the existence of a "program work" in the sense of the Copyright Act. It found that "the IBF file is nothing more than a file containing installation information ... to be installed into the MENU. EXE program when EO System installs each application software into a hard disc. It is not a set of instructions for a computer nor does it act as a program causing the computer to install applications software. In other words, the recorded contents of the IBF file are to be installed into EO System as data, so we must say that the IBF file is simply a data file."[80]

The question is whether it is appropriate to look at the IBF file separately or as some scholars suggest[81] as a unit of MENU. EXE file and IBF file. If it is possible to write the same EO System within one file, such a unit has to be considered. Otherwise the programmer's decision whether to have one big file or several small ones may decide upon whether this program is protected by copyright law. This arbitrariness would be inequitable. Thus it would have been appropriate to regard the IBF file in connection with the MENU. EXE file as a program work.

---

[78] Text of Tokyo High Court Decision on Software Protection, 1 Int'l Computer Lawyer, Aug. 1993, at 32.
[79] Keiji Sugiyama, Reverse Engineering and Other Issues on Software Protection, speech at Computer Law Congress 1991, p.5.
[80] Id.
[81] Sommer, Die Schutzfähigkeit von Computerprogrammen nach japanischem Recht, GRUR Int. 1994, 383, 386.

This does not mean that the file is protected by copyright. As pointed out in the discussion about interfaces, it is not the program as a whole which has to be considered but the several units of which it is composed have to be examined as to whether they are creative. The selection of these units is the most difficult task. If the selected unit is too small, it is normally not an expression but an idea, if the unit becomes too big, there is a variety of expressions, thus the whole unit is protected. In the given case it seems appropriate to regard an IBF file as one unit in order to examine whether it is a creative expression. As the range of commands is very limited this has to be taken as not having the required creative expression. Thus the District Court's decision was correct.

E. Future Protection

As seen before there is still a set of unsettled problems under the present legislation. Therefore it is suitable to think about a more advanced system of protection of computer software.

I. The Piracy Problem

First it must be stressed that the main problem from the economic point of view, that is the problem of piracy (i.e. illegal copying), is not a problem of the present copyright protection. Piracy without any doubt constitutes a copyright infringement. The problem of piracy is more a technical one, as copying the program from one disc to another can even be done by persons who don not have special knowledge about computers.

Although there are methods to prevent the use of illegal copies, e.g. so called "dongles"[82], they are expensive and therefore can only be used for specialized software.

Finally this problem has to be solved by a stricter enforcement in case of copyright infringement. In this respect the campaign started by BSA leads into

[82] A chip that is connected via the parallel interface and interacts with the software. If the software cannot interact with the chip, the software is useless. As the chip is a hardware part, it is much more difficult to be copied.

the right direction. BSA is offering a reward of 10,000yen for information leading to piracy litigation, and 100,000yen for appropriate court testimony.[83]

Another initiative was started by the Cultural Affairs Agency. They announced in 1993 that they are planning to restrict individual copying. For this purpose a committee was installed to examine whether amendments to the law, in particular Art. 30 are necessary. The main concern are so called "copy guards" and programs which enable the user to remove these "copy guards".[84] To what extent these measures will be successful has to be reviewed in the future.

II. Examination of Different Systems of Protection

1) Registration Act

Although some commentators discuss whether the Registration Act can be a proper means of protection[85], the nature of this act contains no protective elements. It is a mere administrative regulation which specifies the rules concerning the registration of computer software provided in the Copyright Act. If one wishes to extend the meaning of the Registration Act to a protective measures this will lead to regulation by a law sui generis. Protection by such a law will be discussed later.

2) Patent Law

Patent Law as shown under part C has some importance in protecting inventions which include software. On the one hand patent law has advantages over copyright law. First the patent is granted after examination. Due to this examination the patent is exactly determined. Therefore there are no doubts as to the scope of a certain patent.

On the other hand there are several problems with patent law. First the existence of novelty is difficult to determine. Programs are often developed in a trial and error procedure, which is an ongoing process more important than the original idea. For the same reason it is difficult to determine the state of art.

[83] World Intellectual Property Report 1995, 15.
[84] World Intellectual Property Report 1993, 208.
[85] Sommer, supra, 388.

Secondly the examination of patent applications takes time. The life of software is comparatively short[86], so that at the time the patent is granted the software may already be outdated. In this context updates also bear problems. They endanger the claims made in the application for the original software. At the same time if a patent is claimed for the update as well the upadate application may be rejected because of the original application if it is not a depending invention in the sense of Section 31 (1) Patent Act.

A third problem is the publication of the invention which means that competitors can more easily base their new developments on former programs.

One may think about lowering the requirements for patents to achieve a better protection of software. Tamai[87] for example proposes a change of the patent acts to grant patents to "things or methods of any kind capable of exerting an action onto the external world without the intervention of the mental process". This is supposed to include all computer programs which have an industrial application.

The following reasons are against such a wide scope of patent protection. It has to be considered that a patent gives a 20 year lasting exclusive right. The standards set for granting this exclusive right must therefore be of a kind which do not hinder technical development. Otherwise patent law would protect a monopolization in the field of software. Accordingly the present required standards seem to be appropriate.

Protection by patent law is therefore restricted to main-frames and firmware, i.e. inventions in which software is incorporated into the hardware.

---

[86] As a representative of Microsoft stated in a U.S. Congress Hearing the life of personal computer software used in a trade or business is no greater then 5 years, Tax Treatment of Intangible Assets : Hearings on H.R. 3035 Before the House Committee on Ways and Means, 102nd Cong., 1st Sess. (1991) at 159 as cited in Band/Katoh p.286 footnote 6.

[87] Tamai, Katsuya ; Patentability of Software-Related Inventions ; in SOFTIC, 4th International Symposium... ; 1993, 341.

3) Trade Secret Law

Although it is generally possible to have software protected as a trade secret, it is not possible in relation to software which is sold to the public. In this case the program ceases to be a secret as it has been made public. This is without any doubt also true where only the object code is sold or licensed. Therefore protection by the Trade Secret Act is only available in a limited number of cases. It seems thus not to be an appropriate general means of protection.

4) Contractual Protection

Kitagawa[88] proposes a protection by contract regulations, the copymart system. For this purpose he argues for the introduction of a network through which copyrighted works should be traded. Although it is true that a contractual system (due to the territorial principle of intellectual property rights) provides better protection across borders, several problems do still remain. For example only the partner of the contract can be held liable in case of illegal copying while no protection exists against further copying. Secondly actions can hardly be supported by criminal action. For these reasons a contractual system can only give additional protection. This is, however, not new: Software producers are already using contracts to prohibit certain actions by users (as seen even more than permitted by the law). Kitagawa's proposal thus has its relevance more in other fields than in software protection.

5) Copyright Law

Copyright Law is so far the most important means of software protection in Japan. Its advantages and disadvantages have been shown at different places of this paper and will therefore not be repeated here. The main problem is that some areas, like reverse engineering, require further regulation. This is mainly due to the fact that a copyright is granted upon creation and accordingly the scope of protection cannot be determined before an actual litigation. Whether further regulations concerning software should be made within the Copyright Act is questionable. There is the danger that the Copyright Act becomes

[88] Zentaro Kitagawa, Computer, Digital Technology and Copyright, in : WIPO Worldwide Symposium on the Future of Copyright and Neighbouring Rights 1994, p.115.

overloaded with provisions concerning software, which pushes the originally protected works into the background. There are several differences between other kinds of works protected by copyright, such as books, movies, pictures, on the one side and computer software on the other side. In the case of computer software the economic situation is much more important. Another aspect is the utilitarian character of software. One more fact is that the distinction between idea and expression is extremely difficult to draw in the case of software. A further aspect is that the normal duration of use is much less than the protected 50 years. As to be seen from reports by software producers a protection period of 5 to 7 years would be sufficient. Therefore copyright seems not the 100% appropriate protective means for computer programs.

On the other hand most countries have a protection under copyright law for software. Now also the TRIPS[89] in Art. 10 require member states to grant copyright protection of at least 50 years from the date of creation. Since Japan is a member country to this agreement, the future protection of software cannot be completely independent from copyright law.

6) Law Sui Generis

Naturally a law sui generis gives the widest range for specific regulations. It thereby leaves space for detailed regulations. For example, the discussed questions of interfaces and reverse engineering could be regulated in sufficient detail. The disadvantage of a law sui generis is, that it is not integrated into an established system (like for instance copyright law). It follows, that as far as a problem is not explicitly regulated by any provision, a solution is difficult to find, because there is no system on which a decision of this question can be based.

III. Proposal : A Copyright which is Based on Law Sui Generis

It was made clear, that the Registration Act, the Trade Secret Act, the Patent Act are not a proper basis for the protection of computer software. Consequently copyright and special legislative protection are left. To combine the advantages and to avoid the disadvantages of both systems, the introduction

[89] Agreement on Trade-Related Aspects of Intellectual Property Rights.

of a copyright based law sui generis seems to be the most appropriate solution. The danger of an overload of software provisions slightly inconsistent with provisions concerning other works in the Copyright Act is thereby avoided. International harmonization is not disrupted. The duration of protection can be adjusted to the short life of software products. Finally for new questions there is a well established system of copyright protection modified by the law sui generis.

F. Conclusion

The major means of protection of computer software is the Copyright Act as amended in 1985. For facilitating the proof of copyright a registration system was introduced. While this legal basis is sufficient for prohibiting piracy, for technical reasons the enforcement causes problems.

There are still unclarified points concerning the protection of interfaces and the permissibility of reverse engineering. Interfaces are not to be protected as long as they prevent interoperability. Reverse engineering is permitted as reverse analyzing. The results of this analysis are just to be used to achieve compatibility. Any unnecessary copying must be avoided by using clean room methods.

Concerning microprograms some broad guidelines have to be given by the legislator to which extent several programs can constitute one unit to be regarded as a program work. These broad guidelines then are to be developed through interpretation by the courts.

For the future a more precise regulation should be thought about. It is proposed that the most appropriate solution is to introduce a copyright based law sui generis. Whether this proposal is to be followed or not, for the future it is most important to find a regulation balanced between a sufficient protection which gives incentive to software producers on the one side and avoids monopolies created by overprotection (by prohibiting the development of interoperable products) on the other side.

Further discussion will definitely come. As this field of law is developing almost as fast as the technology it is dealing with, future developments are difficult to foresee.

Postscript

This article was accepted as a master's thesis in the LL. M. program of Kyushu University in 1995. I would like to thank the organizer of this program, Professor Yanagihara for making this publication possible and Professor Kumagai for being a very active and supportive academic advisor. My thanks also go to all the helpful people at SOFTIC for facilitating my research work.

Appendix I

REGIONAL SURVEY : REMEDIES AVAILABLE FOR INFRINGEMENT OF IP RIGHTS IN COMPUTER SOFTWARE

The Business Software Alliance (BSA) has published estimates as to the volume of software piracy around the world and the consequent loss of revenue to its members. Three Asian countries feature in the top ten of dollar losses. In this special regional survey, *IP Asia* looks at the remedies available for combatting infringement of IP rights in computer software.

The BSA's members are : Aldus Corporation, Apple Computer Inc, Autodesk Inc, Intergraph Corporation, Lotus Development Corporation, Microsoft Corporation, Novell Inc, The Santa Cruz Operation and WordPerfect Corporation.

Country	Losses (US$millions)	Piracy (%)	Country	Losses (US$millions)	Piracy (%)
United States	$2,253	35	Egypt	$84	93
Japan	**$1,961**	**80**	Saudi Arabia	$82	90
Germany	$1,584	57	Greece	$81	75
France	$833	66	Belgium	$79	68
Korea	**$646**	**78**	Norway	$59	67
PRC	**$596**	**94**	South Africa	$55	45
United Kingdom	$492	49	United Arab Emirates	$54	99
Spain	$333	88	Israel	$52	81
Brazil	$331	83	**Hong Kong**	**$49**	**66**
Italy	$324	50	Austria	$48	42
Canada	$234	59	Portugal	$47	77
Netherlands	$216	78	Chile	$45	83
Poland	$216	94	Ireland	$45	49
Czech Republic	$185	86	Colombia	$44	71
Taiwan	**$184**	**84**	Finland	$43	67
India	**$165**	**76**	**New Zealand**	**$36**	**69**
Mexico	$156	80	**Singapore**	**$32**	**63**
Turkey	$140	86	Kuwait	$25	98
Thailand	**$133**	**99**	Peru	$25	98
Indonesia	**$122**	**99**	Paraguay	$12	94
Australia	**$117**	**47**	Bolivia	$10	96
Argentina	$112	74	Cyprus	$4	91
Uruguay	$112	74	**Pakistan**	**$3**	**99**
Sweden	$102	54	Panama	$2	80
Malaysia	**$87**	**98**	Nigeria	$1	83

(IP ASIA, 30 November 1994)

Appendix II

(Example 1)

創作年月日登録申請書
(Application Form for Registration of the Date of Creation)

収入印紙
(Revenue stamp)
(¥3,000)

Date : ___ ___ ___

財団法人ソフトウェア情報センター
理事長_____殿
(Director in Chief of the Software Information Center)

1. 著作物の題号
(Title of the program work) : General Management System for Dispensary
2. 登録の原因及びその発生年月日
(Grounds for registration and the date of creation)
平成___年___月___日に創作した。
(The program was created on _____ _____ _____.)
3. 登録の目的
(Object of registration)
創作年月日の登録
(Registration of the date of creation.)
4. 前登録の年月日及び登録番号
(Date and number of the previous registration.)
な し
(None)
5. 申請者
(Applicant)
住 所
(Address) _____
名 称
(Name) _____
代表者
(Representative) _____
代理人
(Agent)
住 所
(Address) _____ 〒
(Zip code) _____
名 前 印
(Name) _____ (Seal) Tel : _____
Fax : _____
6. 添付資料の目録
(List of attached materials)
(1) 著作物の明細書
(Program work specification) : 1
(2) 代表者資格証明書
(Representative qualification certificate) : 1
(3) 委任状
(Letter of proxy) : 1
(4) プログラムの著作物の複製物
(Copy of the program work) 1
(5) 登録手数料納付書
(Statement of registration fee payment) : 1

Statistics on Program Registration

April 30, 1996

SOFTIC (Software Information Center, Japan)

1. Types of registration

Type	FY1987	1988	1989	1990	1991	1992	1993	1994	1995	1996 (April)	Total
Registration of the date of creation	473	456	521	531	510	542	554	505	489	28	4609
Registration of the date of first publication	26	14	17	5	4	9	5	12	2	1	95
Registration of the true name	4	1	1	1	1	1	1	3	0	0	13
Registration of copyright (in case of copyright transfer, etc)	30	28	42	36	38	48	42	52	41	3	360
Total (*)	533	499	581	573	553	600	602	572	532	32	5077

2. Categories of program

Type	FY1987	1988	1989	1990	1991	1992	1993	1994	1995	1996 (April)	Total
System program	154	100	173	179	111	122	101	112	86	10	1148
General purpose application program	140	156	192	196	193	176	210	172	198	8	1641
Special purpose application program	227	234	213	184	228	281	276	258	236	13	2150
Total (*)	521	490	578	559	532	579	587	542	520	31	4939

(*) The total number of registration exceeds the total number of the categories of program, because a program can be registered in different types.

3. Registration from abroad

Country / Region	FY1987	1988	1989	1990	1991	1992	1993	1994	1995	1996 (April)	Total
U.S.A	1	0	6	2	0	0	1	2	1	0	13
United Kingdom	0	1	0	0	1	1	0	0	0	0	3
Germany	0	3	0	0	0	0	0	0	0	0	3
Spain	0	0	1	0	0	0	0	0	0	0	1
Taiwan	0	0	5	0	0	0	0	0	0	0	5
Israel	0	0	1	0	0	0	1	0	0	0	2
Korea	0	0	0	1	0	0	1	1	2	0	5
Switzerland	0	0	0	0	0	1	0	0	0	0	1
China	0	0	0	0	0	0	1	0	0	0	1
France	0	0	0	0	0	0	0	0	1	0	1
Total	1	4	13	3	1	2	4	3	4	0	35

Bibliography

Band, Jonathan ; Katho, Masanobu ; Interfaces on Trial, 1995

Doi, Teruo ; Computer Technology and Copyright Legislative and Judicial Developments ; Intellectual Property Protection and Management, 1992

Heath, Christopher ; Keine Änderungen des japanischen Urhebergesetzes zum reverse engineering ; Mitteilungen der Deutsch-Japanischen Juristenvereinigung Nr. 13/14, 61

Heymann, Thomas ; The International Effect of the EU Restrictions on Reverse Engineering, The International Computer Lawyer 1994, 15

Hirakawa, Osamu ; *Nakano, Kenichi* ; Copyright of Computer "Interfaces" in Japan, EIPR 1990, 46-50

Karjala ; Dennis S. ; Japanese Courts Interpret the "Algorithm" Limitation on the Copyright Protection of Computer Programs ; EIPR 1990, 235

Kitagawa, Zentaro ; Comment on A Manifesto Concerning Legal Protection of Computerprograms ; Columbia Law Review 1994, 2610

Kitagawa, Zentaro ; Computers, Digital Technology and Copyright ; WIPO Worldwide Symposium on the Future Copyright and Neighbouring Rights

Lehmann, Michael ; The European Directive on the Protection of Computer Programs ; in : A Handbook of European Software Law, 1993, 165

Nakayama, Nobuhiro ; Legal Protection of Software, 1988

Negishi, Akira ; Schutz von Computerprogrammen ; in : Baum/Drobnig ; Japanisches Handels- und Wirtschaftsrecht

Ozaki ; Copyright Protection of Software : The Japanese View, 1990 Computer L. Rep. 950

Pilny, Karheinz ; Schnittstellen in Computerprogrammen - Zum Rechtsschutz in Deutschland, den USA und Japan ; GRUR Int. 1990, 431

Pilny, Karlheinz ; Japanisches Urheberrecht - Entwicklungen und aktuelle Tendenzen ; in : Menkhaus, Das Japanische im Japanischem Recht

Pilny, Karlheinz ; Die Registrierung von Computer Software in Japan ; GRUR Int. 1988, 26

Rahn, Guntrahm ; Sonderrechtschutz für Computerprogramme in Japan ; GRUR Int. 1984, 219

Software Information Center (SOFTIC); Guide to Program Registration

Software Information Center (SOFTIC); 4 th International Symposium on Legal

Protection of Computer Software, November 1993

Sommer, Robert ; Die Schutzfähigkeit von Computerprogrammen nach japanischem Recht ; GRUR Int. 1994, 383

Sugiyama, Keiji ; Reverse Engineering and other Issues on Software Protection ; World Computer Congress 1991

Tanikawa, Hisashi; Standard Form Contract ; in : The Japanese Legal System

White, Ron ; How Computers Work, 1993

Yoshida, Masao ; Creativity and the Scope of Protection of Computer Programs