Washington International Law Journal

Volume 2 | Number 2

7-1-1993

Recent Developments in Copyright Protection for Computer Software in the United States and Japan

Yutaka Nakamura

RECENT DEVELOPMENTS IN COPYRIGHT PROTECTION FOR COMPUTER SOFTWARE IN THE UNITED STATES AND JAPAN

Yutaka Nakamura†

Abstract: Many current legal issues pertaining to copyright of computer software involve defining the scope of protection of non-literal expression, such as "user interface" and "look and feel," in contrast to literal expression, such as "source code," to which ownership may be more clearly attributed. Superficially, it appears that the case law pertaining to non-literal expression is developing differently in Japan and the United States. This comment demonstrates that, however, while Japanese and U.S. courts have been applying formally different analytical criteria, the decisions of both have been similar in seeking equity-oriented solutions.

I. INTRODUCTION

The development of copyright protection for computer software has recently entered a "second generation."[1] When the issue of legal protection for computer software first arose in the 1970s and early 1980s, the discussion centered on whether computer software was protected under copyright. In the United States and Japan, this "first generation" issue was solved by judicial decisions which recognized that literal aspects of computer software, which include the machine-readable "object code" and the programmer-readable "source code," were protected as "literary work" under copyright laws.[2] These holdings were later codified by amendments to the Copyright Act of each country.

In the context of rapid developments in computer technology and computer software markets, the center of the discussion has now shifted to a

---

† B.L., 1986 Kyūshū University; LL.M., 1992 University of Washington.

This article is based on the author's paper presented at the LL.M. Asian Law Program at the University of Washington School of Law. The author wishes to thank Toshiko Takenaka, Assistant Director, Center for Advanced Study and Research on Intellectual Property; Michael Townsend, Assistant Professor of the University of Washington School of Law; and David Lowe and Carl Marquardt for comments on an early draft. The author also owes thanks to Naoki Koizumi, Assistant Professor of Kobe University, and Takehiko Ando for invaluable information on Japanese materials. The author is solely responsible for the shortcomings of this work.

1 *See generally* Edward Samuels, *"The Idea-Expression Dichotomy in Copyright Law,"* 56 TENN. L. REV. 321, 355-366 (1989) for a categorization of the developments of computer software cases. *See also* Menell, *"An Analysis of the Scope of Copyright Protection for Application Programs,"* 41 STAN. L. REV. 1045, 1048 (1989).

2 17 U.S.C.S. § 101 (Law. Co-op. 1993).

"second generation" issue: to what extent non-literal aspects of computer software are protected under copyright.[3] The protectable elements at issue include "structure, sequence, and organization" ("SSO"), "look and feel," "user interface,"[4] and other visual aspects generated by computer software.

At first glance, developments in case law in the U.S. and Japan indicate substantial differences between the two countries. The U.S. courts have relied on common law copyright doctrines, while the Japanese courts have depended on the statutory requirements of copyright. Most noticeably, the U.S. courts have gradually broadened the scope of protection to include non-literal aspects by application of the idea-expression doctrine; the Japanese courts, on the other hand, have yet to recognize non-literal aspects as protectable.[5] This difference is said to be mainly derived from a unique statutory provision of the Japanese Copyright Act which expressly excludes "algorithms" from copyright protection.[6] This view, in conjunction with the conservativeness of the Japanese courts in developing the law, appears to imply that there is little room for the Japanese courts to go beyond the current limited protection as long as the statutory exclusion exists.[7]

---

[3] *See* CRAIG JOICE, ET AL., COPYRIGHT LAW 137 (2d ed. 1991) ["The 'second generation' software cases are concerned with thrashing out *which* elements of protected computer programs should benefit from protection, and *to what extent*" (emphasis original)]. One commentator divides the developments into three "waves": the first wave which dealt with protection for literal codes of computer software, the second wave with "structure, sequence and organization," and the third wave with user interface issues. David Bender, *Computer Associates v. Altai: Rationality Prevails*, 9 COMPUTER LAW. 1 (Aug. 1992).

[4] There is an uncertainty in the definitions of "look and feel" and "user interface." "Look and feel" is generally defined as "a set of functional capabilities of a programmed computer and the way it 'interacts' with a user." Steven Lundberg, et al., *The Copyright/Patent Interface: Why Utilitarian "Look and Feel" is Uncopyrightable Subject Matter*, 6 COMPUTER LAW. 5 (Jan. 1989). Likewise, "user interface" generally refers to "all of the devices by which the human user can interact with the computer in order to accomplish the tasks of the computer program." Joseph T. Verdesca, *Copyrighting the User Interface: Too Much Protection?*, 45 SW. L.J. 1047, 1047 n.4 (1991).

Some commentators and courts have used the "user interface" and "look and feel" of computer software interchangeably. *See, e.g.*, Alan S. Middleton, *A Thousand Clones: The Scope of Copyright Protection in the "Look and Feel" of Computer Programs*, 63 WASH. L. REV. 195, 195 & n.2 (1988); *Lotus Development Corp. v. Paperback Software International*, 740 F. Supp. 37, 63 (D.Mass. 1990). But one commentator states that the majority of scholars differentiate between "look and feel" and "user interface," understanding that "user interface" is one element of "look and feel." Brian Johnson, *An Analysis of the Copyrightability of the "Look and Feel" of a Computer Program: Lotus v. Paperback Software*, 52 OHIO ST. L.J. 947, 953-61 (1991).

[5] For discussion of "literal similarity and non-fragmented comprehensive similarity," *see* MELVILLE B. NIMMER AND DAVID NIMMER, 3 NIMMER ON COPYRIGHT §13.03[A] (1988) [hereinafter NIMMER]. *See also* M. LEAFTER, UNDERSTANDING COPYRIGHT LAW § 9.5, at 269 (1989).

[6] *See, e.g.*, Dennis S. Karjala, *Copyright Protection of Computer Software in the United States and Japan: Part II*, 7 EUR. INTELL. PROP. REV. 231 (1991).

[7] *See, e.g.*, Rovert R. Devéza, *Legal Protection of Computer Software in Major Industrial Countries: A Survey of Copyright and Patent Protection for Computer Software*, 9 PAC. BASIN L.J. 166, 179 (1991). Karjala, *Japanese Courts Interpret the "Algorithm" Limitation on the Copyright Protection of Computer Programs*, 31 JURIMETRICS J. 233, 234 (1991).

This article reviews the recent "second generation" case law in the United States and Japan, with attention to the factual patterns as well as development of legal tests. With this emphasis, the article attempts to demonstrate that superficial differences in the scope of protection under the U.S. and the Japanese case law are a reflection of the different fact-patterns which the U.S. and Japanese courts have faced. This article also demonstrates that the apparent difference in the legal tests applied to determine copyrightability is merely a matter of legal formality; that in fact, the U.S. and Japanese courts have adopted virtually identical processes of analysis with respect to second generation issues.

II. U.S. LAW

A. Background

1. Legislative History of the Copyright Act

The first Copyright Act was enacted in 1790, pursuant to the constitutional authority of the Congress "to Promote the Progress of Science and Useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries."[8] As significant changes in technology occurred, the list of copyrightable work under the Act was gradually broadened by amendments. Consequently, the 1976 Copyright Act defined copyrightable works as "original works of authorship fixed in any tangible medium of expression" and expressly included literary, musical, dramatic, pantomime, choreographic, pictorial, graphic, sculptural, audiovisual, and architectural works and sound recordings.[9]

Although the 1976 Act did not explicitly refer to computer software, it was deemed to fall within the category of "literary works."[10] Recognizing the necessity for special consideration of the problems surrounding the

---

[8] U.S. Const. Art. I, §8, cl. 8.

[9] The Copyright Act section 102(a). The U.S. Copyright Office had already accepted approximately two thousand copyright registrations of computer software as "books" before the 1976 Act on the condition that it was published and contained sufficient original authorship, and submitted to the Office in human-readable form. *See* Samuels, *supra* note 1, at 356 n.156. *See also* Devéza, *supra* note 7, at 180.

[10] *See* Apple Computer Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3rd Cir. 1983). The court examined the legislative history of the section 102(a) of the 1976 Copyright Act, holding that object code of computer programs is protected as literary work. *See also* H.R. Rep. No. 1476, 94th Cong., 2d Sess. 54, *reprinted in* 1976 U.S.C.C.A.N. 5659, 5667 ("The term 'literary works' . . . includes computer data bases and computer programs to the extent that they incorporate authorship in the programmer's expression of the original ideas, as distinguished from the ideas themselves.").

copyright law and computer software, Congress created The National Commission on New Technological Uses of Copyrighted Works (CONTU) to address the problem. In 1979, CONTU recommended that a definition of "computer program" be included in the Copyright Act. In 1980, Congress adopted the recommendation, defining "computer program" in section 101 as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."

2. *The Idea-Expression Doctrine and the Related Concepts*

Under the Copyright Act, computer software is subject to the limitations of protection that commonly apply to all kinds of copyrightable works. The most important rule developed by courts to limit the scope of the copyright protection is the idea-expression doctrine, which holds that copyright protection extends only to the "expression" of an idea and never to the "idea" itself. This doctrine is embodied in section 102(b) of the 1976 Act: "[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery."[11]

This rule was first established in *Baker v. Selden*,[12] in which the court denied copyright protection for the plaintiff's book-keeping system on the ground that to give an exclusive right to the underlying idea itself would allow for a monopoly on ideas which should properly belong in the public domain. The court further stated:

> [W]here the art it teaches cannot be used without employing the methods and diagrams used to illustrate the book, or such as are similar to them, such methods and diagrams are to be considered as necessary incidents to the art, and given therewith to the public.[13]

---

[11] The legislative history refers to the distinction between ideas and expressions in terms of computer software: "Some concern has been expressed lest copyright in computer programs should extend protection to the methodology or processes adopted by the programmer, rather than merely to the "writing" expressing his ideas. Section 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law." *Id.* at 5670.

In Kern River Gas Transmission Co. v. Coastal Corp., 899 F.2d 1458 (5th Cir. 1990), the court described the purpose of section 102(b) as "[i]n drawing this fundamental distinction, Congress balanced the competing concerns of providing incentive to authors to create and of fostering competition in such creativity...."

[12] 101 U.S. 99, 25 L. Ed. 841 (1879).

[13] *Id.* at 103.

This part of the court's opinion represents the concept known as the merger doctrine. That is, where only one or a limited number of ways of expression exist, the expression itself is not copyrightable. The rationale for this rule is that if such expressions were copyrightable, no one could express the underlying idea without infringing on the copyright. This would allow the copyright owner a monopoly on ideas without the limitations imposed by patent law.[14] The merger doctrine thus gives greater force to the idea-expression doctrine.[15]

Like the merger doctrine, the *scenes a faire* doctrine represents an aspect of the idea-expression dichotomy. Under the *scenes a faire* doctrine, an expression that is indispensable or natural to an idea in works dealing with similar subjects is not copyrightable.[16] Although this concept originally derived from the non-copyrightability of stock characters and features of dramatic works, it is now applied to standard or common expressions or features in a variety of works.[17]

Since computer software is typically utilitarian, these doctrines which restrict utilitarian ideas from copyright protection play an important role in determining the scope of protection for computer software.[18] As the U.S. cases discussed below indicate, however, these doctrines have not always provided clear guidelines for courts to distinguish between protectable and unprotectable elements of works at issue. Since these doctrines are stated in quite general terms, they are inevitably difficult to apply in a predictable manner.[19] U.S. case law developments are, in a way, the consequence of the courts' struggle to use these doctrines to establish the appropriate line between protectable and unprotectable elements of a given work.

---

14 *See* Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971): "When the 'idea' and its 'expression' are thus inseparable, copying the 'expression' will not be barred, since protecting the 'expression' in such circumstances would confer a monopoly of the 'idea' upon the copyright owner free of the conditions and limitations imposed by the patent law."

15 *See* Kern River, 899 F.2d at 1463: "The doctrine of 'merger' developed in an effort to deal with this difficulty in locating the precise boundary between idea and expression. The doctrine holds that when the expression of an idea is inseparable from the idea itself, the expression and idea merge."

16 *See* Atari, Inc. v. North American Philips Consumer Electronics Corp., 672 F.2d 607, 616 (7th Cir. 1982) ("*[S]cenes a faire* refers to 'incidents, characters or settings which are as a practical matter indispensable, or at least standard, in the treatment of a given topic.' (quoting Alexander v. Haley, 460 F. Supp. 40, 45), Such stock literary devices are not protectible by copyright."). *See also* Data East USA, Inc. v. Epyx, Inc., 862 F.2d 204, 208 (9th Cir. 1988) ("Nor can copyright protection be afforded to elements of expression that necessary . . . to 'scenes a faire'.").

17 *See* Apple Computer, Inc. v. Microsoft Corporation, 799 F. Supp. 1006, 1021 (N.D.Cal. 1992).

18 *See* Menell, *supra* note 1, at 1103.

19 *See* Samuels, *supra* note 1, at 324.

B. The Development of Legal Tests

1. Early SSO Cases

Early attempts to establish the copyrightability of non-literal software elements resulted in conflict and discord. The conflicting understanding of U.S. courts is well represented by two early software cases; *Synercom Technology Inc. v. University Computing Company*[20] and *Whelan Associates v. Jaslow Dental Laboratory, Inc.*[21]

In *Synercom*, the court denied copyright protection for the SSO of an input format on the ground that they constituted unprotectable ideas. The *Synercom* court concluded that the input format is not an expression separable from the underlying idea, using the analogy of automobile's "H" pattern stick shift.[22] The court reasoned that while copyright protected any mode of expression of the H-pattern, such as a manual, diagram, or photograph, another manufacturer should not be prevented from applying the H-pattern itself in designing a car.[23] This conclusion reflects the policy consideration that such structures or patterns should properly belong to the public domain.[24] Given the conflicting interests between the copyright owner and a competing newcomer in a computer software market, the *Synercom* court opted for a rule enhancing the dissemination of valuable ideas. As a result, the *Synercom* court interpreted the idea-expression distinction to allow little protection for non-literal program elements.

In *Whelan* the court put forth completely different considerations in holding that the SSO of a dental laboratory record keeping program were copyrightable.[25] The central issue for the *Whelan* court was whether it could find infringement based upon the substantial similarity in the SSO of the software, even though the two programs were not identical in terms of literal codes. The court considered the idea-expression distinction as declared in *Baker v. Selden*, and stated that "the purpose or function of a utilitarian work would be the work's idea, and everything that is not

---

20 462 F.Supp 1003 (N.D. Tex. 1978).

21 797 F.2d 1222 (3rd Cir. 1986).

22 *Synercom* at 1013.

23 *Id.*

24 *Id.*

25 Prior to the *Whelan* decision, the court in SAS Inst., Inc. v. S & H Computer Sys., Inc., 605 F. Supp. 816 (M.D. Tenn. 1985) found copyright infringement based on the similarity in the structure of two programs. In SAS, while the court did not articulate the structure of the plaintiff's program, one can also find a strong equity factor in that the defendant had access to the plaintiff's source code in developing the program in violation of a license agreement between the parties.

necessary to that purpose or function would be part of the expression of the idea."[26] Applying this rule, the court found that the idea of the plaintiff's program was "to aid in the business operations of a dental laboratory."[27] The SSO of the program fell within the "everything else" and was therefore copyrightable expression.[28] In so holding, the *Whelan* court rejected the *Synercom* holding that the line between literal and non-literal elements was the appropriate boundary for copyright protection.[29] Instead, it sought to draw a new line between the utilitarian purpose of the program and the expression, with emphasis on the economic incentives for programmers that underlay copyright law policy.[30] The court stated "[t]he rule proposed here, which allows copyright protection beyond the literal computer code, would provide the proper incentive for programmers by protecting their most valuable efforts, while not giving them a stranglehold over the development of new computer devices that accomplish the same end."[31] At this point, the *Whelan* court stood for the opposite interest—the economic interest for authors—which copyright law attempts to balance with public interest emphasized by the *Synercom* court.

---

26 *Whelan*, 797 F.2d at 1222, 1236. This part of the court reasoning has been criticized by courts and many commentators. *See* Apple Computer Inc. v. Microsoft Corp., 799 F. Supp. 1006, 1025 (N.D.Cal. 1992) ("[T]he Whelan rule distends copyright protection, placing off-limits alternative and improved means of expression and thereby upsetting the uneasy balance which copyright attempts to maintain by preventing free riders from ripping off creative expression while not stifling others from improving or extending that expression."); Computer Associates International Inc. v. Altai Inc., 982 F.2d 693, 705 (2nd Cir. 1992) ("Whelan's general formulation that a program's overall purpose equates with the program's idea is descriptively inadequate."); NIMMER §13.03[F], 13-78.33 ("The critical flaw in this reasoning is that it assumes that only one "idea," in copyright terms, underlies any computer program, and that once a separable idea can be identified, everything else must be expression.").

27 *Whelan*, 797 F.2d at 1236.

28 *Id.* at 1238.

29 The *Whelan* court first distinguished *Synercom* on the ground that there was the complexity difference between the input format and computer program itself. *Id.* at 1239. The court also pointed out the *Synercom* court's erroneous understanding of the legislative history of the Copyright Act, stating that Congress did not intend to differentiate the treatment between computer software and other literal works of which arrangement and coordination of the materials are protected under Article 103. *Id.* at 1239-40. Answering the *Synercom* court's question regarding the underlying idea if the structure is expression, the court further emphasized that the idea in plaintiff's program in *Whelan* is the efficient organization of a dental laboratory. Thus, program structures are separable from the idea since there are many possible program structures through which the idea can be expressed. *Id.* at 1240.

30 The *Whelan* court did consider the objective program parts such as file structure and screen output in reaching this conclusion. But the court's discussion is rather limited to the evaluation of a testimony. *See Whelan*, 797 F.2d. 1242-43.

31 *Id.* at 1237. As to the value of non-literal aspects of the program at issue, the court also stated that: "The evidence in this case shows that Ms. Whelan spent a tremendous amount of time studying Jaslow Labs, organizing the modules and subroutines for the Dentalab program, and working out the data arrangements, and a comparatively small amount of time actually coding the Dentalab program." *Id.* at 1231.

In *Broderbund Software, Inc. v. Unison World Inc.*,[32] the court expressly followed *Whelan's* rule. The primary issue argued in *Broderbund* was the copyrightability of a screen display generated by the computer software. The defendant developed and marketed a printing program for IBM machines that had a very similar screen command menu to the plaintiff's successful printing program for Apple computers.[33] In finding the screen display copyrightable, the *Broderbund* court simply applied *Whelan* and recognized the single utilitarian purpose of the work as an unprotectable idea, without verifying the scope of the rule with respect to screen displays.[34] In assuming that the *Whelan* rule applied to computer screen displays in general, the *Broderbund* court did not discuss whether each particular element of the screen display qualified for copyright protection.

After *Broderbund*, however, the Fifth Circuit expressly declined to follow the *Whelan* decision on the copyrightability of SSO in *Plains Cotton Co-Op. v. Goodpasture Computer Service Inc.*[35] Following the *Synercom* decision, the *Plains Cotton* court concluded that the SSO of the plaintiff's cotton market information program was an unprotectable idea, since the SSO was dictated by external factors of the cotton market.[36] While the court did not fully articulate the rationale of the idea-expression doctrine, it embraced the *Synercom* analogy, equating the SSO of the plaintiff's program with the H-shift pattern.[37]

One characteristic of these early cases is that the courts did not undertake much analysis of the non-literal elements of the program at issue when they determined the copyrightability question. The structure and organization of a particular program may have various aspects corresponding to the various operations which the program is intended to perform. In order to clarify the subject matter of copyright in cases

---

32 648 F. Supp. 1127 (N.D.Cal. 1986).

33 *Id.* at 1130.

34 The Broderbund court erroneously interpreted *Whelan* as extending the copyright protection to the "overall structure of a program, including its audiovisual displays." *Id.* at 1133. ("Whelan thus stands for the proposition that copyright protection is not limited to the literal aspects of a computer program, but rather that it extends to the overall structure of a program, including its audiovisual displays."). The *Whelan* court dealt with the issue of the screen display in the context that the similarity of the screen display can be indirect evidence of the decision of substantial similarity between the programs, not as the subject of the copyright protection. (797 F.2d at 1244). *See* Johnson, *supra* note 4, 965 n.88.

35 807 F.2d 1256 (5th Cir. 1987).

36 *Id.* at 1262.

37 *Id.* ("To the extent that input formats represent a level of computer software design more specific than functional design and more general than line-by-line program design, the issue of their copyrightability is relevant to the issue of whether GEMS infringes on protected Telcot designs."). *See* notes 22-24 and the accompanying text.

involving non-literal program elements, an objective analysis of the various aspects of SSO, such as system architecture, algorithms and data structure, is indispensable.[38] These cases, however, seem to stand on generalized concepts of SSO and ambiguous analyses as to which non-literal aspects of a particular program are protected under copyright. Such reasoning, without the reinforcement of objective analysis of SSO elements, resulted in the inconsistent conclusions of these cases. Moreover, it left the impression that rationales of the cases were simply methods for justifying the desired outcome.[39]

2. *Later Screen Display Cases*

Such insufficient attention to program elements was somewhat overcome in later cases dealing with the copyrightability of the visual aspects of computer software. In *Digital Communications v. Softklone Distributing*[40] and *Manufacturers Technologies, Inc. v. CAMS, Inc.*,[41] the courts developed a method of analysis which focused more on the objective characteristics of each element of the work at issue.

In *Softklone*, the plaintiff alleged that the defendant violated the copyright of its successful communication program by copying its well-designed "status screen," which displayed the status of operating parameters and a list of command terms for the user's convenience.[42] In *CAMS*, the defendant developed a cost-estimating program that had a function and design significantly similar to the plaintiff's program.[43]

The notable point in both the *Softklone* and *CAMS* decisions is that the courts overcame the *Whelan* holding that a program can have only one

---

38 One commentator strongly advocates the necessity for establishing such definitions of program parts for orderly development of copyright protection for computer software. John W. L. Ogilvie, *Note: Defining Computer Program Parts Under Learned Hand's Abstraction Test in Software Copyright Infringement Cases*, 91 MICH L. REV. 526 (1992). Likewise, another commentator pointed out that the U.S. courts' lack of objective techniques for separating unprotectable elements ("systems and methods") from a computer program in question has lead to inconsistent case law. Steven W. Lundberg, et al., *Identifying Uncopyrightable Computer Implemented Processes and Systems*, 9 COMPUTER LAW. 7 (Apr. 1992).

39 *See* Samuels, *supra* note 1, 324 ("it seems to be an ex post facto characterization that justifies an outcome based upon other, more concrete factors. Thus, if the outcome in a particular case is to be infringement, the work is deemed to be protectable expression; if the outcome is to be non infringement, then the work is described as an idea.").

40 659 F. Supp. 449 (N.D.Ga. 1987).

41 706 F. Supp. 984 (D.Conn. 1989).

42 *Softklone* at 449. The court followed this registration of the Copyright Office and held that the status screen is considered a compilation of program terms. *Id.* at 462-63.

43 *CAMS*, 706 F. Supp. at 988.

unprotectable idea. Instead, the courts based their opinions on extensive analysis on the objective features of the works. For example, the *Softklone* court identified three elements of the program as unprotectable ideas and then excluded them from the consideration of substantial similarity.[44] The *CAMS* court's analysis of the elements of screen display was even more extensive. The court classified the screen display on a number of levels, such as the sequence and flow of the screens, the screen display format, the method of internal navigation on the screens, and the selection and arrangement of status information as to the operation of the program.[45] In *CAMS*, the analysis of the copyrightability of screen displays is specifically developed based on the classification of the elements of the work. Instead of identifying only one utilitarian "idea" as the court had in *Whelan*, the court identified several ideas corresponding to each element. In this manner, the court could efficiently determine whether each particular element of expression merged with the corresponding idea.

It is also interesting to note that both courts supported the copyrightability of the arrangement, selection, and design of materials in their conclusions. The *Softklone* court held that the arrangement, highlighting and two-letter capitalization of the command menu were copyrightable.[46] Similarly, the *CAMS* court held that the two elements of the screen display relating to the arrangement and selection of materials were copyrightable expressions.[47] Both decisions were supported by the

---

[44] *Softklone*, 659 F. Supp. at 459. The court correctly described that the *Whelan* court did not extend the copyright protection to the screen displays, and rejected the holding of *Broderbund* as being based on misinterpretation of *Whelan*. *Id.* at 455. "[T]he *Broderbund* court based its conclusion on what this court believes to be an overexpansive and erroneous reading of *Whelan* . . . . The Third Circuit in *Whelan* dealt only with the evidentiary use of the copying of screen displays for the purpose of establishing copying of the underlying computer program. The *Whelan* case did not stand for, as *Broderbund* believed, the proposition that screen displays are protected by the computer program's copyright from copying." The court failed to distinguish between the idea of the program itself and the ideas underlying screen displays that the court specified as subject matter of this case. The three ideas that the court enumerated were: (1) the use of the screen display to inform user of the status of the program; (2) the use of a command driven program; and (3) the use of two-key combinations to activate commands. These ideas belong to the general ideas of the program rather than the ideas of the specific screen display at issue. This identification of idea appears to be inconsistent with the court's approach which considers the screen displays as a work separate from the program.

[45] *CAMS*, 706 F. Supp. at 994-99.

[46] *Softklone*, 659 F. Supp. at 463.

[47] Three such elements are: (1) the external sequence and flow of the screen displays in the creating-an-estimate sequence; (2) the selection and arrangement of the status information on the status screen; and (3) the selection and arrangement of the terms on the job identification screen. *Softklone*, 659 F. Supp., 994-96.

analogy to the copyrightability of a compilation work, recognizing the degree of creativity embodied in the original compilation of materials.[48]

In one sense, *Softklone* and *CAMS* represent a turning point in the development of computer software cases. Perhaps as the result of the increased complexity of the programs involved, the courts appear to have gradually moved away from simple reliance on the *Whelan* rule. Instead, the courts are working to establish an analytic "process" for applying the idea-expression doctrine, based on an extensive analysis of the objective factors of the work. In this manner, the courts may reach reasonable conclusions in ever more intricate computer software cases.

3. *User Interface Cases*

The approach to second-generation questions reached a notable stage when the courts considered the copyrightability of "look and feel" and "user interface"[49] in *Telemarketing Resources v. Symantec Corp.*[50] and *Lotus Development Corp. v. Paperback Software International.*[51] In *Symantec*, the court adopted the objective analysis methodology developed by the courts in *Softklone* and *CAMS*. The *Symantec* court held that copyright protection could extend to the "look and feel" of screen displays in general, although the court concluded that no copyrightable element existed in the plaintiff's program.[52]

---

48 *Softklone*, 659 F. Supp. at 462-63. *CAMS*, 706 F. Supp. at 996.

49 "User interface" is one of the non-literal aspects of computer software through which the user communicates with the program. The importance of user interfaces has been increasing in software development since it directly relates to the user's ease of use and productivity, which are significant factors of software competitiveness. *See* Gerard J. Lewis Jr., *Lotus Development Corp. v. Paperback Software International: Broad Copyright Protection for User Interfaces Ignores the Software Industry's Trend toward Standardization*, 52 U. PITT. L. REV. 689, 694 (1991).

50 12 U.S.P.Q.2d (BNA) 1991 (N.D.Cal. 1989). On April 7, 1992, the Ninth Circuit Court of Appeals affirmed the portion of the district court judgment on the copyright infringement claim, while it remanded the case to the district court on the ground that the district court did not decide on the plaintiff's Federal Trademark/Trade Dress claim. *Brown Bag Software v. Symantec Corp.*, 960 F.2d 1465 (9th Cir. 1992).

51 *Lotus*, 740 F. Supp. at 37 (D.Mass. 1990).

52 The usage of the "look and feel" concept in the computer software case has been criticized by several commentators and courts. *See* 3 NIMMER, 13-36 ("it serves no purpose in the realm of computers where analytic dissection and expert testimony emphatically are needed."). In *Autoskill Inc. v. National Educational Support Systems, Inc.*, 793 F. Supp. 1557 (D.N.M. 1992), the court stated to the same effect, citing NIMMER. 793 F. Supp. 1570. *See also Lotus*, 740 F. Supp. at 62 ("Despite its widespread use in public discourse on the copyrightability of non-literal elements of computer programs, I have not found the 'look and feel' concept, standing alone, to be significantly helpful in distinguishing between non-literal elements of a computer program that are copyrightable and those that are not.").

In its landmark holding that the menu command structure of the plaintiff's spreadsheet program was copyrightable, the *Lotus* court first declared a "legal test" for determining whether the user interface at issue was copyrightable. The plaintiff developed a spreadsheet program called "1-2-3," which was designed to meet the need for a more powerful spreadsheet than the first successful commercial computer spreadsheet, VisiCalc.[53] The plaintiff's program had a powerful "user interface" that was different from VisiCalc: a two-line menu command, user-defined function keys and a macro instruction feature.[54] The defendant also developed a spreadsheet program in response to the success of VisiCalc; however, the defendant later decided to improve the program so as to be compatible with 1-2-3 in terms of the user interfaces in order to effectively compete with the 1-2-3.[55] The *Lotus* court set out a three-step "evaluative" test for applying the idea-expression distinction to computer software: (1) the court must determine where to locate the idea underlying the work, between the generalized idea and the particularized expression;[56] (2) the court must determine whether there is a limited number of ways to express that idea or whether the expression is not essential to the idea; (3) if the court finds that the expression is not essential to the idea, it must then determine whether that expression is a substantial part of the work.[57]

Despite the systematic appearance of the evaluative test, the *Lotus* court's conclusion still seemed to be led by the equitable considerations that the non-literal aspects were the most valuable part of the work and that the plaintiff had expended tremendous effort in their development.[58] The *Lotus*

---

53 *Lotus*, 740 F. Supp. at 66.

54 *Id.* at 63-65.

55 *Id.* at 69.

56 In applying the first test, the court held that the idea underlying the work was "the electric spreadsheet" and that such an idea was not copyrightable. *Id.* at 65. The problem with this conclusion is that the basis for the court's finding was not the idea of user interface on which the court should have focused, but merely the general idea underlying the program itself. At this point, the court made a critical mistake, which the *CAMS* court pointed out as a possible mistake in the court's identifying the idea of user interface. Therefore, even though the *Lotus* court properly announced the legal tests, it failed to follow the *CAMS* effective approach in applying the first test for identifying idea. Consequently, this flaw results in the *Lotus* court's holding that the structure of menu commands is copyrightable without identifying what is the specific underlying idea of the structure of menu commands.

57 *Lotus*, 740 F. Supp. at 60-61.

58 *See Lotus*, 740 F. Supp. at 68 ("The user interface of 1-2-3 is its most unique element, and is the aspect that has made 1-2-3 so popular."). Also, the court recognized that the legislative intent of the copyright law mandated more protection for "the intellectual effort and creativity embodied in a user interface" than that under the trade secret law. *Id.* at 56. The *Lotus* courts spent a substantial part of the decision on the discussion of copyright policy issue—economic incentive for authors versus the dissemination of cultural knowledge. In this discussion, the court seems to opt for the economic incentive underlying copyright law, emphasizing the innovative user interface of the plaintiff's program. *See Lotus*, 740 F.

court thus stood on the same policy basis on which *Whelan* court had expanded the copyright protection to SSO. This basis has been criticized in recent copyright cases,[59] leaving uncertainty about the applicability of the *Lotus* test to future user interface cases.[60]

4. *Recent SSO Cases–Filtering Approach*

As new cases have raised increasingly complex issues, courts have gradually shifted the center of analysis to an extensive observation of the elements and structure of the software at issue. In these cases, it is becoming obvious that mere general analysis of the idea-expression doctrine to separate protectable expressions from unprotectable ideas does not provide a useful tool for the courts to determine the scope of copyright protection for various non-literal aspects of computer software. Thus, instead of exploring the limitations of the idea-expression doctrine, recent courts have tried to establish a systematic analytic process for the idea-expression distinction. One such attempt can be seen in the introduction of the "filtering approach" in recent cases, which was originally proposed by Professor Nimmer in his treatise.[61]

In *Autoskill, Inc. v. National Educational Support Systems, Inc.*,[62] the District Court for the District of New Mexico explicitly applied the "filtering approach" to determine the copyrightability of the SSO of an

---

Supp. at 79 ("[T]he more innovative the expression of an idea is, the more important is copyright protection for that expression.").

59 *See infra* note 87.

60 *See* Richard H. Stern, *Legal Protection of Screen Displays and Other User Interface for Computers: A Problem in Balancing Incentives for Creation Against Need for Free Access to the Utilitarian*, 14 COLUM.-VLA J.L. & ARTS 285, 328. "[T]he court's approach raises serious problems because the scope of it is so unpredictable.".

61 *See* Nimmer, et al., *A Structured Approach to Analyzing the Substantial Similarity of Computer Software in Copyright Infringement Cases*, 20 ARIZ. ST. L.J. 625 (1988). *See also* 3 NIMMER, 13-78.28 to .30. "The crucial consideration in the analysis that follows is that copyright law protects only an author's original expression, not just an overall similarity between the works. Thus, before evaluating substantial similarity, it is necessary to eliminate from consideration those elements of a program that are not protected by copyright. To accomplish this task an allegedly infringed program should be analyzed on several different levels. A different copyright doctrine is applied at each level, and material which is unprotected under that doctrine is excluded from further consideration in analyzing substantial similarity. By successively filtering out unprotected material, a core of protected material remains against which the court can compare the allegedly infringing program (Footnotes omitted)." Professor Nimmer further proposed four tests and doctrines excluding program elements that: (1) constitute only abstract ideas; (2) are dictated by logic and efficiency; (3) are dictated by external considerations, such as software and hardware standards and programming practices, and (4) are taken from the public domain. *Id.* at 13-78.30 to .44.

62 *Autoskill*, 793 F. Supp. 1557 (D.N.M. 1992).

educational program for testing and training students.[63] The court refused to adopt the *Whelan* rule on the ground that it neglected the reality that a program may contain more than one idea.[64] Instead, the court adopted the "filtering approach," which allows courts to specify protectable elements by successively applying the tests and copyright doctrines that function to exclude unprotectable materials from the expressions at issue.

As the first step in the filtering approach, the court identified the idea underlying the plaintiff's program: the testing and training of categories of students by use of a computer.[65] The court then successively applied copyright doctrines to the expressions of these identified ideas to exclude unprotectable elements of the expressions. Consequently, it held that the merger doctrine precluded the use of thirteen skill categories based on English vowels and consonants from protection because such categories are dictated by the nature of the English language. It next held that *scenes a faire* doctrine filtered out the use of "Silent Sentence" and "Silent Paragraph" techniques since they are standard methods in education programs.[66] The court then concluded that particular methods of reading and testing employed in the plaintiff's program, such as testing components of sub-types, use of matrices and graphs, were copyrightable expression.[67]

On the other hand, in *Computer Associates International Inc. v. Altai Inc.*,[68] the Second Circuit Court of Appeals denied copyright protection for SSO using the same filtering approach. In *Altai*, the court concluded that the rewritten version of the defendant's program, a compatibility component program that connected the plaintiff's job scheduling program with three different operating systems of the IBM mainframe computers, did not constitute copyright infringement.[69] After rejecting the *Whelan* rule,[70] the

---

63 *Id.* at 1559.

64 *Id.* at 1566. The court described the *Whelan*'s rule as "a temptingly simplistic and bright line test." *Id.*

65 *Id.* at 1566.

66 *Id.* at 1567-68.

67 *Id.* at 1568.

68 *Altai,* 982 F.2d at 693.

69 The district court of *Altai* compared the elements of the two programs' literal codes, parameter lists, macros, the list of services, and high-level structure. First, as to the parameter lists and macros, the court found that many of them were dictated by external factors, that is, the IBM's operating software that belongs to the public domain, and thus not protected. *Altai,* 775 F. Supp. 544, 561 (E.D.N.Y. 1991). Second, with regard to the list of services, while the evidence showed that the two programs shared common services, the court held that they did not constitute copyrightable elements since most of them were determined by the functionality of the program. *Id.* at 562. Finally, as for the high-level structure, the district court held that it constituted so obvious an expression that it could not be protected under copyright. *Id.*

court adopted the successive filtering approach, which it described as a "three step procedure:" Abstraction—Filtration—Comparison. The court found this approach adequate for determining the copyrightability of non-literal elements of a program.[71] Under this approach, the court stated, a court should first break down the allegedly infringed program into several elements.[72] At the first step of abstraction, the court must examine the program and apply the abstraction scale in the order opposite to that which a programmer usually takes during program development—starting with the literal codes and ending with the ultimate function of the program.[73] Once the unprotectable ideas are thus identified, the inquiry moves to the next step: filtering out all unprotectable elements from the overall elements by successively applying the established copyright doctrines such as the non-copyrightability of the elements taken from the public domain or dictated by efficiency and external factors, the merger doctrine, and the *scenes a faire* doctrine.[74] What ultimately remains after application of this test is a "core of protectable material."[75]

Strictly speaking, the filtering approach is a method to allow courts to effectively identify the protectable elements, but it is not a test that automatically separates unprotectable ideas from allegedly copyrightable expressions by its application. Since the filtering approach does not contain any standards that positively ratify the copyrightability of the remaining elements of expression, a court's conclusion that the remaining elements are protectible cannot be ratified unless it has completely filtered out all unprotectable elements by thoroughly applying all appropriate tests and doctrines supported by the given facts in a case. Especially, as Professor Nimmer himself admitted, the first step in the application of the abstraction

---

70 The district court rejected *Whelan* on two grounds. First, it found *Whelan*'s test, based on the proposition that there exists only a separable idea in the computer program, was inadequate since it neglected the reality of the computer program, citing Professor Nimmer's criticism of *Whelan*. *Altai*, 775 F. Supp. at 559. Second, it also found that *Whelan* failed to recognize the fact that a computer program has two distinct aspects: a textual aspect, which includes the source code and object code; and a behavioral aspect, that includes the operations of a program. Both of which have their own structure, sequence and organization. *Id.* at 559-60.

71 *Altai*, 982 F.2d at 706. In finding this, however, the court also noted that this method was subject to change in response to technological development. *Id.* ("[I]n cases where the technology in question does not allow for a literal application of the procedure we outline below, our opinion should not be read to foreclose the district courts of our circuit from utilizing a modified version.").

72 *Id.*

73 *Id.* at 707.

74 *Id.*

75 *Id.*

test would not be a simple task for courts.[76] On this point, for example, the *Altai* court merely explained the general procedure of the approach, but did not proceed to indicate how the court actually identified ideas by applying the abstraction test to the given facts in this case. The *Autoskill* court, on the other hand, still fell into a critical confusion of the general idea of the work and the idea of each particular element at issue in applying the various copyright doctrines to each element.[77] It is conceptually true that the filtering approach can serve as a significant tool for courts in identifying protectable non-literal elements. But the filtering approach is not fully functional until the largely undefined processes of identifying the underlying ideas though the application of the abstraction test are sufficiently articulated. The *Autoskill* and *Altai* courts appear to have left this issue unexplored.

5. *Summary of the Development of Legal Tests*

In the U.S. cases, the courts have been struggling to apply the axiomatic but ambiguous idea-expression dichotomy to the non-literal aspects of computer software. The first notable attempt made by the *Whelan* court resulted in a simple rule for distinguishing ideas from expression, but this holding has been criticized for its extreme expansion of copyright protection based on the erroneous understanding that only one idea exists in a program.

The courts after *Whelan* were required to untangle a more complicated mixture of protectable and unprotectable elements, as well as to analyze the rationale of two conflicting precedents—*Whelan* and *Synercom*—as applied to a series of screen display cases. One consistent tendency that the courts have shown in these cases is their favorable attitude toward the copyrightability of the arrangement, selection, and design of

---

[76] *See* 3 NIMMER, 13-78.33 ("Unfortunately, even in the realm of computer software, the abstraction test is not easy to apply.").

[77] For instance, the *Autoskill* court merely identified the general idea underlying the program: the testing, diagnosing and training of a particular category of students using the program. Consequently, the court analyzed remaining elements after the screening and found substantial similarity because of (1) the existence of a visual scanning test, (2) use of matrices for recording student progress, and (3) use of graphs in the two programs. However, these three elements seem to be in the realm of specific ideas underlying each component of the program. Instead, the court should have questioned the description of such elements employed by the two programs: what style or method of visual tests the two programs employed, or what kind or style of matrix and graphs they used, and considered the substantial similarity in these descriptive aspects. If the court had identified the ideas corresponding to each element of the program component, the court might have found that these two elements should be screened out in the course of successive filtering.

materials.[78] Perhaps because these elements are likely to contain a relatively higher degree of creativity than other program elements, it would be easier for courts to distinguish these elements from other stock expressions or expressions that have limited variations and recognize them as copyrightable expressions.

Because of the *Whelan/Synercom* antagonism, recent courts have abandoned the search for a simple rule to distinguish ideas and expression in computer software; instead they tend to seek an analytic method that enables the courts to efficiently separate the protectable elements from the non-protectable elements in a given case. This approach is now represented by the "successive filtering" approach. While the adaptation of the "successive filtering" approach should prevail on future courts, the current use of this approach by the courts indicates that the process of identifying the underlying idea is not yet adequately defined. Considering the importance of this first step of identifying the ideas, the next issue which will confront the courts following this approach is to clarify the steps or rules for application of the abstraction test, recognizing that computer software contains many ideas corresponding to the numbers of elements or features included in the software.[79]

D. The Factual Situations in the U.S. Cases

Looking to the facts of the U.S. cases in which the courts developed these legal tests, one notices that the courts have mostly reached reasonable and straightforward results in each case, in light of the given facts. The courts have consistently appreciated the equities in the factual situations, and these underlying equitable factors have undoubtedly played a significant role in determining the outcomes of these cases.[80]

---

78 Copyrightability of compilation work consisting of non-protectable materials in other copyright cases would support this trend. *See* Harper House Inc. v. Thomas Nelson, Inc., 889 F.2d 197 (9th Cir. 1989) (schedule organizer), Levine v. McDonald's Corp., 735 F. Supp. 92 (S.D.N.Y. 1990) (music). *See* Michael B. Bixby, *Synthesis and Originality in Computer Screen Displays and User Interfaces: The "Look and Feel" Cases,* 27 WILLIAMETTE L. REV. 31, 49 & n.112 (1991).

79 The difficulty that the courts have had in identifying the underlying idea of a program in the application of the idea-expression doctrine can be seen in other cases prior to the filtering approach cases. See the previous discussions of *Softklone, supra* note 44; *Lotus, supra* note 56; *Autoskill, supra* note 77.

80 Some commentators pointed out that the existence of equity in computer software cases has significant effects on the decisions. *See, e.g.,* Richard D. Moreno, *"Look and Feel" as a Copyrightable Element: The Legacy of Whelan v. Jaslow? Or, Can Equity in Computer Program Infringement Case Be Found Instead By the Proper Allocation of Burden of Persuasion?* 51 LA L.REV. 177 (1990). *See also* Bixby, *supra* note 78, at 43. Dennis S. Karjala & Keiji Sugiyama, JAPAN-U.S. COMPUTER COPYRIGHT LAW 196 (in Japanese, 1989).

For example, most of the cases in which the courts protected non-literal aspects involved virtually identical copying of some critical elements: in *Broderbund*, there was "the eerie resemblance between the screens of the two programs;"[81] in *Softklone*, stylistic expressions embodied in the two programs were virtually identical.[82] In a more recent case, the *Lotus* court also recognized that the user interface of the two programs was "strikingly similar." The primary copyright law policy of enhancing knowledge does not favor this sort of "verbatim copying," since the second author does not make any creative contribution or investment in the work.[83]

The courts have also been influenced by "bad faith" copying in cases where the parties have had a close business relationship. In *Whelan*, the defendant had received substantial technical assistance from the plaintiff, which enabled the defendant to develop its own program, and moreover, the defendant was a sales representative of the plaintiff's program.[84] The *Whelan* court implied that the defendant could not have completed the development of the program without the plaintiff's extensive efforts.[85] In *Broderbund*, the court appeared to be influenced by the equitable considerations that the plaintiff had previously disclosed to the defendant the source code and technical points of the program in the course of license negotiations.[86] Likewise, the defendant in *CAMS* was working as a sales representative for the plaintiff's program and apparently developed its program based on information obtained through this relationship.[87] Such

---

81 *Broderbund*, 648 F. Supp. at 1137. One commentator described *Broderbund* as "actually [an] equity case, in which the court was so outraged by the extent of copying that the court was going to find an infringement one way or the other." Johnson, *supra* note 4, 965 [citing Conley, *'Look and Feel' in Defense of the Current Case Law*, 5 COMPUTER LAW. 1, 2 (Dec. 1988)].

82 *See* 659 F. Supp. 449, app. Exhibit A and B at 465-66. They apparently indicate that the two screens are identical, except for the lists of available commands. On this point, *Softklone* is distinguishable from *Synercom* because the input format consisted of a mere sequence of data that contained minimal stylistic creativity.

83 *See Apple*, 799 F. Supp. 1006, 1021-22.

84 *Whelan*, 797 F.2d at 1226-27.

85 The *Whelan* court acknowledged some other factors that support the defendant's "free-riding" activities. For example, the defendant failed to develop the program by himself due to his inexperience in computer software prior to receiving help from the plaintiff. *Whelan*, 797 F.2d at 1225. The district court of *Whelan* acknowledged that the defendant was quite familiar with the plaintiff's program and he utilized the source code of the plaintiff's program in his attempt to develop the IBM-PC Dentcom program. *Whelan*, 609 F. Supp. at 1321. The district court also considered the fact that the defendant stated in the advertisement of his program, "At last the 'dentalab' system available under ten thousand dollars." *Id.* at 1322.

86 While the license negotiation consequently failed, the defendant apparently utilized such information in developing his program. *Broderbund*, 648 F. Supp. at 1130.

87 *CAMS*, 706 F. Supp. at 988. The access of the defendants in *CAMS* was limited to the appearance of screen displays, and promotional material of the plaintiff's program. The court expressly denied that the defendant had access to the source code of the program. *Id.* at 988-89. While the plaintiff's claim based on

situations seem to move the courts to extend copyright protection in order to achieve justice in the case at hand.

These factual findings were sometimes supported by the courts' recognition that a particular non-literal aspect may be the most valuable product of the author's efforts.[88] The *Whelan* court expressly noted that the plaintiff's valuable efforts to develop the program's SSO should be protected under copyright; otherwise useful technology developments would be hampered.[89] In *Lotus*, the court also appeared to appreciate the fact that it was the user interface of the plaintiff's program which made the program so popular.[90]

Thus, the expansion of copyright protection to non-literal aspects was to some degree propelled by strong equitable factors in the factual situations. The courts found it necessary to preclude "bad faith" free riders who misappropriated the most valuable portions of programs. This contrasts well with the factual situations in *Plains Cotton*, *Symantec*, and *Altai*, the cases in which courts denied copyright protection for non-literal elements. In these cases, the "good faith" defendants did not engage in verbatim copying.[91] In *Plains Cotton* and *Altai*, the courts appear to have appreciated

---

the breach of fiduciary duty came from this business relationship, the court rejected this part of the claim on the ground that the plaintiff failed to establish that it had control over the defendant's activities. *Id.* at 1005.

88 This proposition, what is known as the "sweat of the brow" doctrine (a concept that copyright is a reward for the author's hard efforts) was flatly rejected by a recent U.S. Supreme Court decision ("[T]he primary objective of copyright is not to reward the labor of authors."). Feist Publications Co., Inc. v. Rural Telephone Service Co., Inc., 111 S. Ct. 1282, 1290 (1991). Shortly after the *Whelan* court supported this doctrine, the same Second Circuit rejected the "sweat of the brow" doctrine in Financial Information, Inc. v. Moody's Investors Service, Inc., 808 F.2d 204 (2d Cir. 1986): "[C]opyright is not to be determined by the amount of effort the author expends, but rather by the nature of the final result." *Id.* at 207. The Second Circuit in *Altai* also rejected the sweat of the brow doctrine in computer software cases, expressly referring to *Feist* and *Whelan*. *See Altai*, 982 F.2d at 706. *See also Autoskill*, 793 F. Supp. at 1571

89 *See supra* note 30 and accompanying text.

90 *See supra* note 56 and accompanying text.

91 In *Symantec*, seventeen alleged similarities did not include virtually identical elements. Also, *Symantec* involved a unique factual situation in which two programs that were to perform the same function were developed by the same programmer. It is thus natural that the two programs share common features. That is not unusual in the software industry. Even in such a situation, the programmer may not be precluded from applying the same idea to another program that he or she may develop in the future. Such a preclusion would amount to an unreasonable limitation on the programmer's own skill and experience. In most computer software cases, the policy issue of the idea-expression distinction is presented as a determination of the parts of an existing program (developed by another programmer) a programmer may use and which parts the programmer may not. In *Symantec*, however, the policy issue involved a more serious and complicated phase; that is, which parts of a program (developed by a programmer but assigned to a third party) the programmer may use repeatedly in other programs as ideas constituting the programmer's own skill or experience, and which parts the programmer may not use. The same issue was presented in a music case which has a similar factual situation as *Symantec*. *See* Fantasy Inc. v. Fogerty, 654 F. Supp. 1129 (N.D. Cal. 1987).

the fact that the defendants made reasonable efforts to avoid misappropriation of the plaintiffs' programs in developing or rewriting their own programs.[92] While these two tendencies will be interwoven in future developments of the case law, it appears that in the future courts will become more careful in expanding copyright protection to non-literal aspects through the development of the filtering approach.

III. JAPANESE LAW

A. *Background*

Until the early 1980s, legal protection for computer software in Japan was a fairly non-existent issue because there were few disputes concerning computer software. However, after a series of cases involving popular video games, the issue was recognized as a problem which called for legislative action. With respect to statutory protection of computer software, there was a heated debate between two government offices: the Ministry of International Trade and Industry (MITI), which administers Japanese patent policy, and the Cultural Affairs Agency (CAA), which administers Japanese copyright policy. MITI advocated the creation of a *sui generis* right for computer software, arguing that this approach would provide legal protection appropriate for its status as economic property.[93] On the other hand, CAA's proposal was based on the facts that computer software was already held by Japanese courts to be copyrightable, and that the international trend was toward protection of computer software under copyright law. CAA thus proposed amending the Copyright Act in order to clarify the scope of the protection for computer software under the statute.[94]

In 1985, the Japanese Diet finally adopted the CAA's proposal in order to be in harmony with copyright protection systems of the other

---

92 For example, the *Plains Cotton* court found that: (1) the defendants developed their program based on "their knowledge of the cotton industry and expertise in computer programming and design gained over a number of years;" (2) the defendant's employer replaced the copied subroutine from the plaintiff's program and discharged the responsible employee; and (3) one of the defendants wrote subroutines without referring to any material or information concerning the plaintiff's program. *Plains Cotton*, 807 F.2d at 1259-61.

93 The MITI proposal emphasized the creation of a right of use (*Shiyo-Ken*). The proposal also contained such notable features as: (1) the abolition of moral rights; (2) a registration requirement for legal protection; (3) a 15-year term, in parallel with patent law; and (4) an introduction of compulsory licensing. *See Proposal with Respect to Rearrangement of the Foundation for Software–Aiming at Security Legal Protection for Software* (in Japanese, MITI, Dec. 1983).

94 *Interim Report of the Copyright Council*, Sixth Subcommittee (in Japanese, CAA, Jan., 1984).

leading countries.[95] Under the new Copyright Act, computer software is protected as a separate category, "program works" (*Programu no Chosakubutsu*). Article 2(1)(x) now defines program works as "an expression of a combination of instructions and statements capable of causing a computer having information processing capabilities to achieve a particular result," and Article 10 includes program works in the enumeration of works of authorship [Article 10(1)(ix)].[96]

Now that computer software is expressly covered by the Japanese Copyright Act, it of course must satisfy the general requirement for works of authorship[97] in order to qualify for protection. Article 2(1)(i), which defines the works protected by copyright, states that a protectable work: (1) must express thoughts or sentiments; (2) must be expressed in a creative way; (3) must fall within a literary, academic, artistic or musical domain.[98] In contrast to the U.S. Copyright Act, which does not have a definition of protected works, the Japanese Copyright Act seems to set higher requirements with respect to the qualities of works eligible for copyright protection.[99] The "creativity"[100] requirement in particular has functioned as

---

95 *Id.* The report states that England, Germany, France and Netherlands were of the opinion that computer software should be protected under the copyright law although none of them had any explicit provisions in their copyright acts. It is also said that the Diet's adaptation of the CAA's proposal was under strong pressure from the U.S., since the U.S. had already established copyright protection for computer software by the 1980 amendment of the Copyright Act. The U.S. opposition to MITI's proposal was especially due to its compulsory licensing provisions that would have allowed Japanese companies to license U.S. companies' software by paying a small amount of royalties. *See* Nobuhiro Nakayama, *Sofutouea no Houteki Hogo*, 12 (Legal Protection for Software) (1988) (in Japanese). However, it appears that the U.S. argument was based on a misunderstanding of the MITI proposal. First, MITI's proposal was limited to only three cases: (1) where a program is made by using existing programs or patented inventions; (2) where compulsory license is necessary for the public benefit; or (3) where the existing programs have not been used, and the proposal explicitly indicated that the provisions must be construed so as not to invade the existing right of the copyright owner. Second, the compulsory license was not automatically allowed, as is the U.S. copyright compulsory license (§ 111, § 115, § 116, § 118, and § 119 of the U.S. Copyright Act), but made upon the authorizing committee's decision. *See id.* at 12-13.

96 The 1986 Amendment also included "database" as deserving a separate category of protection as a kind of compilation work. Article 2(1)(x-iii) defines "database" as "collection of theses, numbers, figures, or other information, which are systematically organized so that they can be searched by using computers." Also, Article 12-1(1) states that a "[d]atabase which has creativity in selection of information or its systematic organization shall be protected as a work of authorship."

97 Under the Japanese Copyright Act, "fixation" is not required for a work in general to be protected. The fixation requirement is expressly provided for cinematograhic works under Article 2(3).

98 Article 2 defines "works of authorship" as "the creative expression of thoughts and sentiments which belongs to the literary, academic, art or music domain." Article 2(1)(i).

99 *See* Dennis S. Karjala & Keiji Sugiyama, *Fundamental Concepts in Japanese and American Copyright Law*, 26 AM. J. COMP. L. 613, 621 (1988).

100 The original Japanese term "*sousakusei*" can be translated to either " originality" or "creativity." However, "creativity" seems to be the better choice so as to distinguish "*sousakusei*" from "originality."

a substantial threshold which the Japanese courts have often invoked to bar copyright protection. Generally speaking, therefore, the creativity requirement under the Japanese Copyright Act is not necessarily equivalent to the "originality" requirement under the U.S. Copyright Act, since the latter is not recognized as a substantial obstacle to copyright protection. In the recent Japanese computer software cases discussed below, the creativity requirement has been utilized to deny copyright protection for program elements when protection would result in a monopoly of the underlying ideas.

One of the most significant outcomes of the 1985 Amendment is that the Japanese Copyright Act expressly provides for exclusions from copyright protection that are only applicable to program works. Article 10(3) states:

> The protection granted by the Law to the works of authorship enumerated in (1)(ix) of this article does not extend to a programming language, rule, or algorithm which is used for creating the works. In this case, the meanings of these terms are subject to the following definitions:
>
> (i) "Programming language" means characters or other symbols or their organization, which are used for methods of expression.
> (ii)"Rule" means special conventions with regard to the usage of the programming language defined in the preceding item in a particular program.
> (iii)"Algorithm" means methods for the combination of instructions to computers.

Among these exclusions, the exclusion of the algorithm has an important role in determining the scope of the protection for non-literal elements of a program. Because the scope of the exclusion is clarified in the statute, Japanese courts supposedly can draw a clearer line between protectable and unprotectable elements in computer software cases without relying upon the ambiguous idea-expression doctrine.[101] However, recent Japanese cases

---

See Karjala, *supra* note 6, 237, n.19. *See also* Edward G. Durney, *Protection of Computer Programs Under Japanese Copyright Law*, 9 PAC. BASIN L.J. 17, 38 n.61 & 40 n.69 (1991).

[101] Unlike the U.S. Copyright Act, the Japanese Copyright Act does not have a provision expressly excluding "ideas" from copyright protection. The general concept of idea-expression distinction has been established by the courts. They have interpreted that the usage of the word "expression" in Article 2(1)(i) excludes ideas from the copyright protection. *See, e.g., Ueda v. K.K. Nishin* (Perpetual Calendar Case),

seem to indicate that there is some uncertainty in the Japanese courts' application of Article 10(3).

B. *The Early Case Law*[102]

1. *The Development of Legal Tests*

The cases categorized as "early case law" below are cases decided based on the Japanese Copyright Act before the 1985 Amendment. Since the previous statute did not clearly establish the copyrightability of computer software, the courts in these cases mostly discussed "first generation" issues, i.e., the question of whether computer software was protected under copyright, and if so, under which category of works. While none of the cases in this category directly discusses the copyrightability of non-literal aspects of computer software, the analysis in these cases and the courts' interpretation of relevant articles of the Japanese Copyright Act provide important clues to the analysis of the recent case law, discussed later.

K.K. Taito v. K.K. ING Enterprises[103] is a landmark case in which a Japanese court first held that the object code of computer software was copyrightable. The plaintiff was the manufacturer of "Space Invader II," a video game which enjoyed great commercial success in this market. The defendant, upon request of customers who owned outdated video game machines, converted those machines to run the plaintiff's video game.[104] In doing so, the defendant installed the object code stored in the plaintiff's ROM chip onto the ROM chips of the customers' machines.[105] In holding that the object code stored in the plaintiff's ROM chip was a copyrightable

---

Judgement of Jan. 24, 1984, Osaka District Ct., 1102 Hanrei Jihō 132. In *Ueda*, the court held that the structure of plaintiff's perpetual calendar was an utilitarian idea that was not protected under copyright. *See also* Karjala & Sugiyama, *supra* note 99, at 649.

102 In Japan, there are two special divisions of district court for dealing with intellectual property law cases; Twenty-ninth Division of Tokyo District Court and Twenty-first Division of Osaka District court. Since each division consists of the judges who are experts in intellectual property law, their decisions are treated to have more precedential value than other courts decisions. Because the judges of each division changes frequently, the decisions from the same division are not always in accord. *See* Karjala & Sugiyama, *supra* note 99, n.6 and accompanying text.

103 Judgement of Dec. 6, 1982, Tokyo Dist. Ct. 1060 Hanrei Jihō 18.

104 *Id.* at 21.

105 *Id.* The defendant argued that the object code of the plaintiff's program was not copyrightable, since such a machine language is neither human-readable nor the expression of human thoughts. The defendant thus argued that it fell outside the categories of copyrightable works enumerated in Article 10(1) of the Copyright Act. *Id.* at 20.

work of authorship,[106] the court reasoned that the object code program could be deemed a creative expression of the author's academic thoughts, since it was based on the author's discovery of solutions that necessitated the author's logical thinking.[107] The court also held that the defendant's conduct in extracting the object code from the plaintiff's ROM chip and installing it onto other ROM chips constituted a copying of the plaintiff's work under Article 2(1)(xv) of the Japanese Copyright Act. Accordingly, the court concluded that the defendant's conduct infringed the plaintiff's copyright.[108]

In *Konami Kogyo K.K. v. K.K. Daiwa*,[109] the court undertook the same logical process as *Taito* in finding the object code of the plaintiff's program was copyrightable. The plaintiff was a manufacturer of a video game "Strategy X." The defendant marketed a game called "Strong X," which was virtually identical to the plaintiff's game. In the manufacturing process, the defendant extracted object code from the plaintiff's ROM on the printed circuit board and copied it onto the defendant's ROM. The plaintiff asserted that its video game was protected under copyright as a literary work. The defendant unsuccessfully contended that the plaintiff's video game could be easily developed by using basic knowledge of video games, and hence that it lacked originality. The court found that the plaintiff's video game program was copyrightable because of its creative expression of thoughts different from other creators. The court also held that the defendant's conduct in extracting the object code from the plaintiff's ROM chip and installing the partially changed code onto other ROM chips constituted a reproduction of the plaintiff's work.[110]

K.K. Namco v. Suishin Kogyo K.K.[111] is the only Japanese case which dealt with the copyrightability of the screen displays generated by computer software. The *Namco* court held that the screen displays were copyrightable as "cinematographic works" before the 1985 Copyright Act. In *Namco*, the plaintiff claimed that defendants infringed the plaintiff's copyright of the video game "PAC-MAN" as a cinematographic work by "performing" the

---

106 *Id.* at 21.

107 *Id.*

108 *Id.* The *Taito* case was later followed by the Yokohama District Court in K.K. Taito v. Makoto Denshi Kogyo K.K, in which the facts were identical. Judgment of Mar. 30, 1983, Yokohama Dist. Ct., 1081 Hanrei Jihō 125.

109 Judgment of Jan. 26, 1984, Osaka Dist. Ct. 536 Hanrei Taimuzu 450.

110 In this case, the plaintiff also asserted that the work was alternatively protected as a cinematographic work, but the court did not address this issue in its decision.

111 Judgment of Sep. 28, 1984, Tokyo Dist. Ct., 534 Hanrei Jihō 246.

game to the public through illicit copies of the game machine.[112] The court first stated that cinematographic works under the Copyright Act must satisfy three requirements: (1) the work must generate visual or audiovisual effects similar to the effects of cinematographic works; (2) it must be fixed in a medium; and (3) it must be a work of authorship.[113] The court found that "PAC-MAN" satisfied those three requirements. The first and third requirements were satisfied without difficulty. The court reasoned that displaying pictures on a video screen has similar effects to cinematography, and that the plaintiff's video game contained creativity as a production of the creator's intellectual or cultural activities.[114] With regard to the second requirement of fixation, the court considered the issue of whether the screen displays of the plaintiff's video game were capable of being reproduced.[115] The court held that they were reproducible, reasoning that if a player operates the game machine in the same way in two different plays of the game, the screen displays varied in the same way.[116]

In *Microsoft Corp. v. Shuwa System Trading K.K*,[117] the court addressed the issue of the copyrightability of operating system software. The plaintiff, an American corporation, developed an operating system program

---

112 The defendants argued that the plaintiff's video game did not fall within the definition of cinematographic works for two reasons. First, it contended that the pictures generated by the video game were mere tools for playing the game like a chesspiece and a chessboard. *Id.* at 251. Second, it argued that the performance rights to cinematographic works were inherent in their unique distribution customs which were utterly different from that of video games. *Id.* at 251-52. The court expressly rejected the second argument reasoning that there already existed other types of works protected as cinematographic works that did not use the same distribution method as cinematographic works. A work is not precluded from the category of cinematographic works by reason of a difference in distribution method. *Id.* at 253-54.

113 *Id.* at 252. The court further elaborated the interpretation of these three requirements. First, with regard to the first requirement, the court concluded that the visual effects were the essential element of cinematographic works. *Id.* at 253. As long as a work has the visual effects that are similar to cinematographic works, it can meet this requirement regardless of the purpose of the work and the interaction of players with the game. As to the second requirement, the court stated that the medium in which the work must be fixed can be anything, including ROM chips. *Id.* The third requirement, the court stated, was further divided into two elements under Article 2(1)(i) under the Copyright Act: (1) the work must consist of creatively expressed thoughts or sentiments, and (2) the work must belong to the domain of literature, science, art, or music. *Id.* The court then construed "literature, science, art, or music" to be the product of intellectual or cultural activities in general. Therefore, whether the work has artistic, utilitarian, or amusement purpose does not affect the copyrightability of the work. *Id.*

114 *Id.* at 254.

115 The Japanese Copyright Act does not generally require "fixation" in order for a work to be copyrightable. Cinematographic work is the only exception which must meet the fixation requirement (Article 2(3) of the Japanese Copyright Act).

116 543 Hanrei Jihō at 254. It is interesting that this reasoning is identical with that in Midway Manufacturing Co. v. Artic International, Inc., 547 F. Supp. 999 (N.D. Ill. 1983), in which the plaintiff, a copyright holder of PAC-MAN in the United States, sued the defendant who made speed-up kits for PAC-MAN.

117 Judgment of Jan. 30, 1987, Tokyo Dist. Ct. 1219 Hanrei Jihō 48.

that worked on the PC-8001 personal computer manufactured by Nihon Electric Company ("NEC").[118] Neither the plaintiff nor NEC disclosed the structure of the operating system program to the public.[119] The defendant, considering the needs of PC-8001 users to know the structure of their operating system program, published a book which listed the source code of the program, together with labels and annotative comments on the list.[120] In the course of making the book, the defendant translated the object code of the program into hexadecimal code and then disassembled and interpreted the object code.[121] The court recognized that the program was created by using a high degree of technical knowledge of the program language, and therefore it was regarded as a work within the academic domain.[122] The defendants contended that an operating system software should not be protected under copyright, since the creator's thoughts and sentiments were excluded from the operating system software for the purpose of more efficient and quick data management.[123] The court rejected this argument, reasoning that an operating system program, like an application program, can have uniqueness by reflecting a creator's character in analyzing the problems to achieve the program's purpose and in selecting an algorithm from various choices.[124] The court then discussed whether disassembling and interpreting the object code and attaching the labels and comments was also a reproduction. While the court acknowledged the differences in expression between the object code and the book, it found that those differences were derived only from the more explanatory purpose of the book, and were therefore unimportant.[125] Accordingly, the court concluded that disassembling and interpreting the object code and attaching the labels and comments also constituted a reproduction of the plaintiff's program.[126]

---

118 *Id.* at 49.
119 *Id.*
120 *Id.* at 52.
121 *Id.*
122 *Id.* at 55.
123 *Id.* at 52.
124 *Id.* at 55.
125 *Id.*
126 *Id.* at 55-56. The court also rejected the defendants' arguments of fair use and non-creativity. The defendants argued that the publication of the book required independent research and creative activities and that it was made for the benefit of NEC PC-8001 users. In rejecting these arguments, the court reasoned that whether or not an allegedly infringing work has creativity or individuality did not affect the determination of infringement, and that the disclosure of the source code of the program to the public against the author's intention was not justified by the fact that it was done for the convenience of the users. *Id. See* Nakayama, *Publication of a Disassembled Operating System*, 634 HANREI TAIMUZU 46, 49 (1987). But one commentator states that the creator's desire to keep the program as a trade secret should not affect

2. *Factual Situations*

As shown in the above cases, early development of legal protection for computer software in Japan was triggered by a series of video game program cases: *Taito*, *Konami*, and *Namco*. Through these three cases, the Japanese courts established copyright protection for computer software, which was later codified in the 1985 Amendment. It should be noted that there is one common factual situation among those three cases; that is, the defendants' programs in these cases were the products of defendant's mechanical or "dead" copying of the literal code of the plaintiff's program. In each case, the fact that the defendant's program was a virtually verbatim copying of the code of the plaintiff's program was not disputed. In other words, as for the copying, the defendants were mere "free riders" contributing no creativity and no cost investment. As such, they could not benefit from copyright law's underlying policy to enhance the culture and learning of the public.

The *Microsoft* case appears to carry a slightly different meaning than these cases. The defendant's book and the plaintiff's program were not identical since they had different modes of expression. However, as the *Microsoft* court recognized, the difference was derived from the difference in the nature of the works: a book and a program. Aside from this difference, the defendant's work was based on an identical copy of the source code of the plaintiff's work. The factual situation of the *Microsoft* case thus can fall in with the verbatim copying cases.

In sum, all of the cases discussed above share a common factual situation, in that each involved virtually verbatim copying of the literal aspects of programs by the defendant. Under such factual circumstances, the only reasonable conclusion was to hold for the plaintiff by affirming the copyright protection for the infringed program. In these cases, none of the courts were faced with the difficulty of drawing a line between the idea and the expression; nor were they required to balance the policy considerations favoring each party.

---

the conclusion. *See* Karjala, *The First Case on Protection of Operating Systems and Reverse Engineering of Programs in Japan*, 10 EUR. INTELL. PROP. REV. 172, 176 & n.27 (1984).

C. *Current Case Law*

1. *The Development of Legal Tests*

In two recent decisions, *System Science K.K. v. Tokyo Sokki K.K.*[127] and *K.K. ICM v. K.K. Met's,*[128] Japanese courts addressed the second generation issue, the copyrightability of non-literal aspects of software. In these provisional relief cases, the courts discussed copyrightability of "processing flow" (*Shori no Nagare*) or organization of a program, which seems to be the equivalent with what the U.S. courts call SSO.[129] While both of the courts rejected the copyrightability of "processing flow" or organization of programs, their analyses conflict in terms of the statutory grounds for why such elements may not be copyrightable.

In *System Science*, the plaintiff alleged that the defendants retrieved four of the plaintiff's programs from the ROMs installed in the plaintiff's hardware systems and copied them onto the ROMs in the defendants' hardware systems.[130] One of the four programs, the CA-9 program, was allegedly an adaptation of the plaintiff's CA-7 II program. Some literal expressions and "processing flow" of the CA-9 were identical with those of the CA-7 II program.[131]

The court upheld the district court's decision that CA-9 did not constitute an infringement of the plaintiff's copyright of CA-7 II.[132] The court first supported the district court's proposition that there must be "creativity" in the similarities between the two programs in order for the infringement to be established.[133] The court then stated:

---

127 Judgment of June 20, 1989, Tokyo High Ct., 1322 Hanrei Jihō 138.

128 Judgment of Mar. 27, 1991, Tokyo Dist. Ct. 23-1 Chiteki Saishu 139.

129 Professor Karjala expressed a similar opinion that the processing flow mentioned in *System Science* "sounds very much like" what the *Whelan* court defined as SSO. Karjala, *supra* note 6, 232-33.

130 1322 Hanrei Jihō at 141-42.

131 *Id.* at 142-43.

132 The district court first discussed whether or not CA-9 was an adaptation of CA-7 II. The court stated two requirements for the infringement of the adaptation right: the access to the plaintiff's program and the similarity between the two programs. It further stated that the infringement of the adaptation right was not established if similarities between the two programs only exist in the portion of the program in which no creativity could be found. *Id.* at 143.

133 1322 Hanrei Jihō at 140. "In order for a program to be found to infringe [the] copyright of a program work, it is required that a portion of the combination of instructions of the program work be found to be creative, and the combination of instructions of the later developed program be similar to the portion of the program work that can be found to be creative."

> In a program, the symbols used for expression are extremely limited, and the system (grammar) of the programming is strict. Because of this, if it is intended to effectively obtain the same result by having a computer operate, there is inevitable room for similarity in the combination of instructions. Therefore, with regard to a computer program, the determination of copyright infringement must be made carefully.[134]

Based on this recognition, the court found that the similarities between the two programs were inevitable because of hardware constraints and common methods of expression. Therefore, these aspects of the program contained no creativity.[135] With regard to the "processing flow" of the program, the court held that it was not protected under copyright since it fell within the definition of "algorithm" (*kaiho*) under Article 10(3)(iii) of the Japanese Copyright Act.[136] Thus, the similarities in this regard were excluded from the determination of creativity. Finally, the court noted that the sizes of the two programs were substantially different; 12K bytes (12,288 bytes) of CA-7 II and 763 bytes of CA-9. The court also noted that the allegedly similar portion of the programs was very small compared to the overall size of the two programs.[137] Accordingly, the court concluded that there was not sufficient evidence showing that CA-9 constituted an infringing adaptation of CA-7 II.

The court's opinion that "processing flow" is unprotectable algorithm is a landmark opinion, but one should be careful in assessing this authority. First, this opinion was stated as dictum. In *System Science*, the center of the appeals court's discussion was on whether there was a lack of creativity in the literal aspects of the program. Second, this portion of the appeals court's opinion was so brief that it did not include any elaboration of the meaning of "processing flow" in connection with the definition of "algorithm" under Article 10(3)(iii), or any explanation of a concrete rationale for equating "processing flow" with "algorithm."[138]

---

134 *Id.*
135 *Id.*
136 *Id.*
137 *Id.*
138 The District Court considered the similarities in the "processing flow" in the analysis of creativity by specifying several examples of "processing flow" at issue. The Tokyo High court, however, did not address this part of the district court opinion. 1322 Hanrei Jihō at 142.

In the most recent case, *K.K. ICM v. K.K. Met's*, the plaintiff developed a utility program that automatically installed application programs onto the hard disks of NEC PC-9801 personal computers.[139] The allegedly infringed program was one of the component files that constituted the plaintiff's utility program, which was called the "IBF file." The IBF file consisted of 81 subfiles, corresponding to each of 81 popular application programs for PC-9801, together with stored instructions for each particular application program.[140] The IBF file was designed to provide such instructions and information for the "MENU.EXE file" so that it could execute the installation process of various application programs.[141]

The defendants developed a utility program that allegedly had the same function, file structure and organization as the plaintiff's program. The defendants' utility program had a file that corresponded to the IBF file, which was called the "HCA file." The plaintiff claimed a copyright infringement of 41 of the 81 IBF files and sought provisional relief to stop the defendants' sales of the program. The plaintiff alleged that the defendants' HCA file was a copy of the IBF file, since there was a one-to-one correspondence with the IBF file in terms of "processing flow" and "algorithm."[142]

Presuming that the IBF file could be regarded as a program, the court concluded that the IBF file was not copyrightable on three grounds. First, the court held that the "organization"[143] of the IBF file was not protected under copyright since it was a grammar (*Bunpou*) of characters, other symbols or their systems under Article 10(3)(i) of Japanese Copyright

---

139 23-1 Chiteki Saishu at 140.

140 *Id.* at 141-42.

141 *Id.* at 153. The structure of the IBF file was not complicated. It consisted of 9 lines such as: the "ID Line," which enables MENU.EXE to check if the file was the correct IBF file; the "Title Line," which describes the name of the application program covered by the file; the "Device Line," which provides information about the device driver; the "Command Line," which creates an auto-execution batch file for the application program; the "Message Line," which contains the message displayed on the screen instructing users to insert floppy disks of the application programs; and the "Installation Process Line," which provides MENU.EXE with directions to copy the designated files onto subdirectories of the hard disk.

142 *Id.* at 144. The defendants raised two arguments against the plaintiff's claim. First, they argued that the IBF file was not a program but a mere data file dictated by simple Japanese and English. Second, they also argued that the IBF file did not meet the statutory requirement of creativity since it was a mere combination of the names of application programs and selection of files that should be installed on the hard disk, which was pre-determined by each application program.

143 It is not clear from the record what the court meant by "organization" of the IBF file. Considering that the court referred to the sequence and structure of the IBF file, e.g. the ID line, the Title line, the Device Line, etc., when it reasoned that such "organization" was not copyrightable, the court seemed to regard what the U.S. court calls SSO as just the "organization" of the file.

Act.[144] The court also held that there was no room for creativity in the selection of forms since the number of ways to select the files was constrained by MENU.EXE.[145] Second, as to the literal expressions of the IBF file, the court held that they also lacked creativity because the lines constituting the IBF file were dictated by the external factors, MENU.EXE[146] and the application program to be installed. Third, as to the copyrightability of the selection of files to be installed on a hard disk, the court held that the selection was an idea not protected under copyright.[147]

The most important aspect of *ICM* is that unlike *System Science*, the court did discuss the copyrightability of non-literal aspects of the program in a way directly relating to the holding of the case. The *ICM* court's treatment of the "organization" of the IBF file in connection with the Article 10(3), however, seems to be inconsistent with the *System Science* analysis. In *ICM*, the court presumably referred to SSO simply as "organization," and held that it fell within the category of "programming language" under Article 10(3)(i). On the other hand, the *System Science* court treated SSO, which it called "processing flow," as "algorithm" under Article 10(3)(iii). While there is ambiguity in the language used by both courts, it appears that *System Science* and *ICM* flatly denied the copyrightability of the SSO on different statutory grounds. Thus, the *ICM* decision at least indicates that Japanese courts' treatment of SSO is not yet unified.[148]

---

144 *Id.* at 157. The court did not provide further rationale for the ambiguous equation of blank form with a grammar of programming language under Article 10(3)(i).

145 *Id.*

146 *Id.* at 157-58.

It is doubtful whether MENU.EXE was a completely external factor to the IBF file. Both the IBF file and MENU.EXE constitute the component parts of one program and they are designed by the same plaintiff. It appears that the plaintiff designed the IBF file as a database for MENU.EXE so that they can interact to efficiently perform the specific function of the program. While the plaintiff did not claim the copyright infringement of MENU.EXE, both files appear to be sufficiently tied with each other so that they could represent the author's creativity as a whole. Moreover, the doctrine of external factors is to preclude copyright protection from a work that comes from considerations "*external to an author's creativity.*" 3 NIMMER 13-78.36 (emphasis added).

Considering those factors, the MENU.EXE could not be truly external to the IBF file since MENU.EXE is in any event within the creativity of the same author of the IBF file. Looking at both files this way, one cannot simply say that the IBF file lacked the creativity on the ground that it is constrained by MENU.EXE. Likewise, one commentator stated that the court should have considered both the IBF file and MENU.EXE as one work of authorship. The commentator also stated that the IBF file could be protected as a kind of compilation work or database file under Article 12 or 12-2 of Japanese Copyright Act. Masao Yoshida, "*Purogramu ni okeru Fairu no Seikaku to Sousakusei*" (*Comment: The Character and Creativity of Files in a Program*), 108 KŌGYŌ SHOYUKENHŌ KENKYU 24 (in Japanese).

147 *ICM*, 23-1 Chiteki Minshu 159.

148 The *ICM* court's equation of SSO with programming language would sound more likely to be erroneous than *System Science* court's equation with algorithm. It is obvious that the definition of "programming language" under Article 10(3)(i) means the method of expressing a program such as BASIC,

2. *Factual Situations*

The *System Science* court was faced with a factual situation which a Japanese court had never before addressed: a work which was not a verbatim copy of the literal code but was identical in non-literal aspects to the plaintiff's program. On this point, the court excluded "processing flow" from consideration and concluded that there was no infringement, even though "processing flow" of the two programs was allegedly identical. The key facts in this opinion are that the sizes of the two programs were substantially different (763 bytes to 12,288 bytes), and moreover, that the size of the allegedly similar portion was a tiny portion of the 763 bytes.[149] These facts indicate not merely that the CA-9 was far from an identical copy of CA-7 II as a whole in terms of literal aspects, but also that the non-literal aspects of such a trivial portion could not raise a substantial similarity issue. If the court had considered the issue of the similarity of "processing flow," it would have held that the similarity was in any event not substantial due to the nominality of the allegedly similar "processing flow."[150] In other words, since the CA-9 was not significantly similar to begin with, the court did not need to enter deeply into consideration of whether the non-literal aspects of the CA-7 II should be copyrightable.

As with *System Science*, the facts in *ICM* with regard to the simplicity of the program at issue provides a clue to understanding the value of this decision. It should be noted that the structure of the file in question was extraordinarily simple for computer software. The IBF file itself contained only 9-lines of structure and was written in mostly plain Japanese and English.[151] Thus the defendant persuasively argued that it could not be a program work. The court, however, did not discuss this issue; instead it assumed that the file was a "program work" and then proceeded to discuss whether it satisfied the statutory requirement of creativity. That is, the *ICM* court's holding concerning the copyrightability of the structure of the IBF file was made without considering whether or not the IBF file itself satisfied the elements of a "program work."[152] The IBF file seemed to be so

---

COBOL and FORTRAN. It appears to be difficult to expand the meaning of "programming language" to something covering the SSO of a program.

149 *See System Science*, 1322 Hanrei Jihō 140.

150 Of course, it is theoretically possible that "processing flow" is qualitatively significant. At best, the plaintiff did not assert that it would represent the critical portion of the program.

151 *See supra* note 141.

152 On March 31, 1992, the Tokyo High Court denied the appeal of the defendant in *ICM*. While the details of the decision are not available, the high court held that the IBF file was merely a data file,

primitive that there could be no legal discussion as to whether its non-literal aspects qualified for copyright protection.

D. Summary and Analysis of Case Law—Comparative Aspects

1. "Originality" vs. "Creativity"

The Japanese case law, when compared to the U.S. case law, seems to indicate that the Japanese courts have avoided the complicated idea-expression doctrines adopted by the U.S. courts. Instead, mainly because the nature of a civil law system makes the court more statute-oriented, Japanese courts have relied heavily on the statutory requirement of creativity in determining the copyrightability of literal aspects of computer software.

On this point, Japanese law contrasts with U.S. law, where in "originality" is a de minimus standard, requiring merely that a work must be an independent creation of author in order to be protected.[153] Thus, the "originality" requirement has been construed not to include positive creativeness such as novelty or ingenuity.[154] On the other hand, the "creativity" requirement under the Japanese copyright law functions as the sole evaluative scale in determining the copyrightability of computer software.[155] The decisiveness of the creativity requirement has been consistent since the *Taito* court held for the first time that a computer program was copyrightable.

---

therefore did not meet the requirement of "program work." In so holding, the high court did not discuss whether the IBF file fell within "programming language" or "algorithms" under Article 10(3) (Interview with Professor Naoki Koizumi on Apr. 2, 1993).

153 *See* H.R. Rep. No. 94-1476, 51-52 (1976), *reprinted in* 1976 U.S.C.C.A.N. 5659: "The phrase 'original works of authorship,' which is purposely left undefined, is intended to incorporate without change the standard of originality established by the courts under the present copyright statute. *This standard does not include requirements of novelty, ingenuity, or aesthetic merit, and there is no intention to enlarge the standard of copyright protection to require them.*" (emphasis added).

154 *See, e.g.*, Alfred Bell & Co. v. Catalda Fine Arts, Inc., 191 F.2d 99, 102 (2d Cir. 1951), ("'Original' in reference to a copyrighted work means that the particular work 'owes its origin' to the 'author.'"); Atari Games. v. Oman, 888 F.2d 878, 883 (D.C. Cir. 1989), ("The level of creativity necessary and sufficient for copyrightability has been described as 'very slight,' 'minimal,' 'modest.").

But see Feist Publications, Inc. v. Rural Telephone Service Company, Inc. 111 S. Ct. 1282 (1991). In *Feist*, the court denied the copyright of the telephone directory on the ground that it lacked minimum originality ("17 U.S.C.§ 101 does not afford protection from copying to a collection of facts that are selected, coordinated, and arranged in a way that utterly lacks originality.").

155 The importance of the creativity requirement under the Japanese copyright law is not only to the computer software cases. Japanese courts have consistently evaluated the degree of the creativity of alleged infringed work as a decisive factor in determining the infringement of subject matter of copyright in general. *See* Takashi Yamamoto, *infra* note 156, 28-32. Shigeru Miki, *The Scope of the Legal Protection for Computer Software*, 98 HŌGAKU SHINPŌ 351, 362-70 (in Japanese).

However, the application of a single evaluative scale in determining copyrightability does not mean that Japanese courts are simply drawing the line between protectable and unprotectable elements without considering as many factors as the U.S. courts do. Rather, as noted earlier, the Japanese courts have evaluated various limiting factors to copyrightability in locating the given work on a continuous creativity scale. For example, in *System Science* the court discussed whether the literal aspects of the CA-7 II were dictated by hardware constraints and common expression; the *ICM* court discussed the external factors that determined the literal expression of the IBF file. If such limiting factors are present, the Japanese courts are likely to conclude that the work does not satisfy the statutory requirement of creativity.

At a glance, the Japanese courts thus set a high standard of copyrightability by requiring a work to be an independent creation of an author, and also to have a substantial quantum of creativity. However, as seen in U.S. cases discussed earlier, the U.S. courts denied the copyrightability of an expression that met the "originality" requirement if it was dictated by such external factors by virtue of copyright doctrines. In Japanese copyright law, there are no independent doctrines or statutory provisions that correspond to the merger, *scene a faire* and common expression doctrines in the U.S. law. Instead, Japanese courts have developed these important copyright law principles within the statutory requirement of creativity.[156] This requirement can ultimately serve the same policy of denying copyright protection if protecting expressions would result in a monopoly of ideas.

In this sense, there seems to be no substantial difference between the U.S. law and the Japanese law with respect to the "difficulty" of recognizing the copyrightability of a literal expression. In Japanese case law, the same analytic process as in the U.S. copyright doctrines are incorporated into the analysis of the quantum of creativity under the Article 2(1)(i) of the Japanese Copyright Act. *System Science* and *ICM* indicate that the Japanese courts appear to be continuing this trend for the time being. In other words, in spite of the difference in formality, both the U.S. and Japanese courts

---

156 Professor Karjala pointed out that the same result as *System Science* can be reached under various U.S. copyright doctrines. Karjala, *supra* note 6, 245. For an excellent analysis of the relationship between the Japanese creativity requirement and the U.S. merger doctrine in connection with *System Science*, *see* Takashi Yamamoto, *Chosakukenho niokeru "Sousakusei" no Gainen to Merge Riron (The Concept of "Creativity" in Copyright Law and Merger Doctrine)*, 456 NBL 27 (1990) (in Japanese). In the article, he also provides precedents in the Japanese copyright law which indicate that the Japanese courts have consistently considered the same principle as the merger doctrine in the creativity requirement.

require basically the same elements in order for a program to be found copyrightable.

2. *Second-Generation Issues*

In contrast, with respect to the "second generation" issue of the protection for non-literal aspects, the approach of the Japanese courts appears to be significantly different from the U.S. approach. Because of the existence of the specific provisions for exclusion from copyright unique to program works, the Japanese courts have tended not to find non-literal aspects copyrightable. However, the attitude of the Japanese courts toward copyright protection for non-literal aspects still contains many uncertain factors.

First, it should be noted that so far only two Japanese cases, *System Science* and *ICM*, mentioned the issue of copyright protection for non-literal aspects of software. These two cases exhibit conflicting understandings of the statutory basis by which "processing flow," which seems equivalent to what the U.S. courts call SSO, could not be copyrightable. In addition, the courts in these cases were reviewing provisional relief and thus working with limited factual records.

Second, and more importantly, the Japanese courts have yet to face the sort of cases that propelled the U.S. courts to expand copyright protection to non-literal aspects in order to effectuate justice in specific cases, i.e., factual situations involving copying of visual aspects without copying the literal code, or business relationships through which a free rider obtains proprietary information. In addition, Japanese courts have yet to address "valuable" non-literal aspects which could lead the courts to expand the scope of copyright protection, as in *Whelan*, *Broderbund*, and *Lotus*. The factual situations which the Japanese courts have encountered mostly included verbatim copying of literal codes, or at best simple programs in which the SSO was too primitive to be worthy of discussion as to whether it should be protected.

This "hidden" but crucial disparity in the factual bases between the U.S. and the Japanese cases results in an apparent difference in the scope of copyright protection in the two countries. In the situations which the Japanese courts have addressed, there has emerged no necessity to reconsider the legitimacy and the scope of Article 10(3) with respect to the protection of non-literal elements. The Japanese courts have yet to ask themselves the fundamental question which the *Whelan* court faced when it

went beyond protection of literal elements: which non-literal aspects of software should be protected under copyright?

VI. CONCLUSION

Now that the general issue of the copyrightability of computer software has been established by statute in both the U.S. and Japan, U.S. and Japanese courts have recently been trying to establish the appropriate scope of protection. Closely observing the cases in the U.S and Japan, one notices interesting similarities in the application of legal tests to second generation issues, even while there have been differences in the factual situations.

The Japanese court's reliance on a statutory requirement of creativity appears to be significantly different from U.S. analyses, which extensively discuss the applicability of various copyright doctrines. However, as demonstrated above, this is a mere difference of legal formality. Because of Japanese courts' general tendency to construct their analysis so as to be faithful to the structure of the statute, they seem to rely on a single statutory requirement. In reality, the statutory requirement of creativity under Japanese copyright law functions as an evaluative scale that includes virtually the same aspects of analysis as the copyright doctrines which the U.S. courts consider in computer software cases.

With respect to the factual differences between the U.S. and Japanese cases, the Japanese cases lacked the sort of equities that compelled the U.S. courts to expand copyright protection to non-literal aspects. In addition to being based on unique copyright law provisions that limit protection for non-literal aspects, the Japanese holdings have been based on facts involving verbatim copying of literal code or primitive SSO which has made it unnecessary to discuss copyright protection for non-literal aspects of programs. This characteristic of the Japanese cases may be partly due to the fact that the Japanese domestic software industry for personal computers is still under development. In addition, there is relatively less recognition among Japanese of the value of computer software as opposed to hardware in general.[157]

The question then arises as to what will happen if a Japanese court in the future faces the sort of situation which the U.S. courts have faced. Is there no room for the Japanese courts to protect non-literal aspects as long as the limitation of Article 10(3) exists? One possibility that the Japanese

---

157 MITI admitted that the weakness of Japanese software industry compared with the hardware industry is the current problem of Japanese computer industry. *See* NIHON KEIZAI SHINBUN, Sep. 29, 1992.

courts have yet to exploit is the copyrightability of "selection, arrangement, and design of materials" as a compilation work, as established by the U.S. courts in *Softklone* and *CAMS* with regard to elements of screen displays. While none of the Japanese cases discussed above addressed this issue, the Japanese Copyright Act explicitly states that a compilation work is copyrightable as long as it meets the creativity requirement.[158] Japanese copyright case law at this point seems to favor the copyrightability of works which demonstrate creativity in the arrangement and selection of materials.[159]

Despite the existence of limiting provisions in the Japanese Copyright Act and Japanese courts' relative lack of judicial creativeness, Japanese courts are attentive to the equities in a given case. There is no apparent difference between the U.S. and the Japanese courts in this regard. If a Japanese court were to face a case like *Lotus*, in which the non-literal aspects of a program have significant value, or a case like *Whelan* or *Broderbund*, where strong equity considerations play a role, how it would respond is uncertain. This comparison of recent U.S. and Japanese cases suggests that the expansion of protection experienced in the United States has not yet come to Japan because of the absence of appropriate cases. Thus, it may be just a matter of time before similar developments occur in Japan.

158 Article 12 of the Japanese Copyright Act provides that: "A compilation work (except for a work which falls within "database") which has creativity in selection or arrangement of materials shall be protected as a work of authorship."

159 *See* Karjala & Sugiyama, supra note 99 nn.131-45 and accompanying text.