The public funding section of House File 200 is directed at the rising costs of publicly funded medical assistance programs. It attempts to control costs while still maintaining a high standard of care for the indigent patient. Eliminating the patient's choice between two or more therapeutically equivalent drug products appears to be a permissible means to achieve this goal.

V. CONCLUSION

House File 200, as enacted by the Iowa Legislature, confers a benefit on the consumer and the pharmacist. Product selection by the pharmacist enables the patient to purchase therapeutically equivalent prescription drugs at a reduced cost. Although the projected savings are only five to eight percent,[75] this is a significant figure when viewed in the context of soaring medical costs. Product selection would particularly benefit patients who are on long term maintenance therapy or those with chronic illnesses.

Product selection also permits the pharmacist to assume a more active role in his professional capacity, utilizing his skill and training in performing a necessary public service. Added privileges carry added burdens, however, and House File 200 is no exception. By becoming an active participant in product selection, the pharmacist must assume greater responsibilities toward the patient. His potential liability is increased, but to such a small degree that it will probably have very little impact on the pharmacist.

The only drawback to House File 200 is the provision regarding public funding. In an attempt to conserve public funds, the legislature passed this ambiguous provision, the effects of which are yet to be determined.

JOHN J. GAJDEL

75. *Pharmacist's Role, supra* note 4, at 195.

PROTECTION OF COMPUTER SOFTWARE— A HARD PROBLEM

The protection of proprietary rights in computer hardware[1] and software[2] has been the subject of much discussion since the advent of extensive usage of automation in the early 1950's. Developers of computer systems are understandably interested in protecting their investments in both hardware and software. Users of computers are similarly concerned with obtaining protection for the various uses to which they put their computer.[3] Protection of computer hardware inventions is readily available in the United States patent system.[4] The problem with which computer developers and users are confronted is protecting the computer programs which make these machines perform. Traditionally, three avenues of protection have been utilized: trade secret; copyright; and patent. None, however, has provided computer software the full scope of protection desired by its owners.[5]

I. Trade Secret Protection

Treatment of computer programs as trade secrets is probably the most obvious means by which their protection may be obtained. Industry practices

---

1. Computer hardware consists of all the machinery which is a part of the computer system. This includes the central processor, input-output control systems, input-output devices (such as tape drives, printers, and disk drives), and various types of memory systems.

2. Computer software is commonly known as computer programs. A program in its simplest form is a set of instructions telling the machinery (hardware) how to perform the particular desired task. A lucid description of this operation is given by Judge Baldwin in *In re Ghiron*, 442 F.2d 985, 986 (C.C.P.A. 1971).

3. A computer user's interest stems from his extensive investment in development of new computer uses. At the outset it simply takes time to conceive of a programming method which will do the required job and to write the program—perhaps seven to ten hours. Additional time is required to thoroughly test the program—usually about twice the programming time. Both programming and testing require computer time and man-hours for operation of the computer. Thus a user can easily spend several hundred dollars for a simple program. If the program or series of programs is complex (which is the usual case), the investment becomes substantial.

4. The machines themselves may be protected as "machines" under 35 U.S.C. § 101 (1970). In addition, the electronics and other integral devices may be protected individually by patents under the same section. The manufacturing of modern electronics may well be protectible as a process—or even as a composition of matter, e.g. the "chips" and printed circuits which are currently in use to provide miniaturization of computers. 35 U.S.C. § 101 (1970).

5. A general discussion of these three methods of protection may be found in articles collected in 3 *Bigelow Computer Law Service*, chapter 4 (1975). *See also* Scafetta, *Computer Software Protection: The Copyright Revision Bill and Alternatives*, 8 J. Mar. J. Prac. and Proc. 381 (1975) [hereinafter cited as Scafetta]; Note, *Computer Software: Beyond the Limits of Existing Proprietary Protection Policy*, 40 Brooklyn L. Rev. 116 (1973) [hereinafter cited as *Beyond the Limits*].

have shown that trade secret law is the most widely used method of protecting software.[6]

Trade secret law is the common law protection of ideas.[7] Ownership of a trade secret gives a proprietor the right "to make use of it to the exclusion of the world."[8] The generally accepted definition of trade secret is contained in the *Restatement of Torts*:

> A trade secret may consist of any formula, device or compilation of information which is used in one's business, and gives him an opportunity to gain an advantage over his competitors who do not know or use it. It may be a formula for a chemical compound, a process of manufacturing, treating or preserving material, a pattern for a machine or other device, or a list of customers.[9]

The *Restatement* implicitly indicates that protection is granted only to limited classes of subject matter.[10] Programs easily satisfy this "subject matter" requirement, since they are processes used in business enabling their owner to gain competitive advantage.[11] Inherent in the nature of the trade secret protection device is an additional requirement that property for which protection is sought to be kept secret.[12] In some instances, trade secret law provides protection even if secrecy is lost; however, a requirement of this protection is an evidenced intent to keep the property secret.[13] It is a relatively small burden on the owner to keep his software secret. Few people are able to read and understand it, and in most instances even this minor danger of disclosure may be avoided by storing software in the computer in a form unreadable even to those few who *can* decipher software in its printed form. Obviously, it is in the owner's interest to keep secret a property which increases his competitive advantage, for if its secrecy is lost, so too is its value to him.[14] Thus, trade secret law is the "natural" protection for computer programs; proprietors simply obey their instincts to keep their worthwhile programs secret.

---

6. Bender, *Trade Secret Protection of Software*, 38 GEO. WASH. L. REV. 909 (1970).
7. *See generally* 2 CALLMANN, UNFAIR COMPETITION, TRADEMARK, AND MONOPOLY, ch. 14 (3d ed. 1968) [hereinafter cited as CALLMANN]; *Developments in the Law—Competitive Torts*, 77 HARV. L. REV. 888, 947 (1964).
8. Hamilton Mfg. Co. v. Tubbs Mfg. Co., 216 F. 401, 407 (6th Cir. 1908).
9. RESTATEMENT OF TORTS, § 757, comment *b* (1939).
10. E.I. DuPont de Nemours Powder Co. v. Masland, 244 U.S. 100 (1917); CALLMANN, *supra* note 7, § 51.1.
11. Courts in several recent decisions have recognized that software may be a trade secret. Telex Corp. v. IBM, 367 F. Supp. 258, 325-26 (N.D. Okla. 1973), antitrust aspects of case reversed but trade secret aspects affirmed, 510 F.2d 894 (10th Cir. 1975) (program used by IBM system engineers for diagnostic and debugging purposes in the initial set-up of a computer system); University Computing Co. v. Lykes-Youngstown Corp., 504 F.2d 518, 534-35 (5th Cir. 1974) (system of programs for retail inventory control); Electronic Data Systems Corp. v. Kinder, 360 F. Supp. 1044, 1048-51 (N.D. Tex. 1973), *aff'd*, 497 F.2d 222 (5th Cir. 1974) (system of programs capable of mechanically processing applications for payment of Medicare claims); Com-Share, Inc. v. Computer Complex, Inc., 338 F. Supp. 1229 (E.D. Mich. 1971), *aff'd*, 458 F.2d 1341 (6th Cir. 1972) (systems software for a time-sharing system).
12. *See, e.g.*, Smith v. Dravo, 203 F.2d 369 (7th Cir. 1953).
13. CALLMANN, *supra* note 7, § 53.1.
14. *See* Coca-Cola Bottling Co. v. Coca Cola Co., 269 F. 796, 805 (D. Del. 1920).

Even though an owner's rights in software guarded as a trade secret are virtually unlimited,[15] this method of protection is not without its disadvantages. A property's trade secret status may be lost if the public obtains knowledge of it.[16] Once the trade secret is published, the secrecy requirement for protection is no longer satisfied.[17] Publication may occur by disclosure in an article in a trade journal or magazine,[18] at a public meeting,[19] or simply through open, unrestricted use within the company.[20]

The latter method of publication is most applicable to computer programs as trade secrets. While it is unlikely that software will be published in trade journals or at a public meeting, many programs are constantly used within a company and readily accessible to employees able to understand software. However, as observed in *Com-Share, Inc. v. Computer Complex, Inc.*,[21] there exist a variety of measures by which a software owner is able to avoid disclosure and maintain trade secret status for his product. Measures validated by the court in that case included: 1) affixing a notice of confidentiality, *e.g.*, "Com-Share, Inc. Company Confidential," on each page of the program; 2) utilizing "password access"[22] to the programs; and 3) keeping print-outs and magnetic tape copies of the programs in a locked compartment when not in use by authorized employees.[23]

In *Telex Corp. v. International Business Machines*,[24] the maintenance of trade secret status was also well illustrated. In an effort to keep confidential the contents of the program, IBM's programmer wrote the complex diagnostic program in source code[25] and then translated it into object code.[26] Since object

---

15. Hamilton Mfg. Co. v. Tubbs Mfg. Co., 216 F. 401 (6th Cir. 1908).
16. CALLMANN, *supra* note 7, § 53.3.
17. Dollac Corp. v. Margon Corp., 164 F. Supp. 41 (D.N.J. 1958).
18. Van Products Co. v. General Welding & Fabricating Co., 419 Pa. 248, 213 A.2d 769 (1965).
19. Nuss v. United States, 117 F. Supp. 413 (Ct. Cl. 1954).
20. U.S. Plywood Corp. v. General Plywood Corp., 370 F.2d 500 (6th Cir. 1966).
21. 338 F. Supp. 1229 (E.D. Mich. 1971).
22. "Password access" is a system which prevents the operation of a program unless the user inputs the correct "password."
23. Com-Share, Inc. v. Computer Complex, Inc., 338 F. Supp. 1229, 1234 (E.D. Mich. 1971).
24. 367 F. Supp. 258 (N.D. Okla. 1973).
25. Source code is typically the first encoded form of a program. A short example of source code is:

```
RD-MASTER.
    READ GL-MASTER-IN AT END GO TO EOJ-MAS.
COMP-COM.
    IF CC-NUMBER LESS THAN GL-COM PERFORM
        RD-COMP-CTRL GO TO COMP-COM.
    IF GL-COM LESS THAN CC-NUMBER GO TO
        RD-MASTER.
    IF CC-GLP15 = 0 GO TO RD-MASTER.
    MOVE CC-MO-END-DATE TO SLASHED-DATE.
    MOVE CORRESPONDING SLASHED-DATE TO WRK-DATE6.
    MOVE R-WRK-DATE TO GL-BALDT.
```

26. Object code is machine-generated code. The source code is read by the computer and compiled into object code, which may actually be used by the computer. A simplified description of the difference between source code and object code is that source code

code cannot be translated to source code,[27] the property was secure so long as the source code form of the program was secure. The court held that IBM had maintained the trade secret status of their program since the object code form of their program was the only form in daily use, the source code copy having been in a locked place.[28]

Although case law has validated measures to insure the secrecy of software, the implementation of such measures is somewhat cumbersome when the software is in constant use.[29] This is one aspect of trade secret protection which makes it less attractive to software owners.

It should be observed that trade secret protection for property may be lost in ways other than through the owner's disclosure. If the property he claims is in the public domain, the owner can claim no rights to it.[30] Similarly, if an alleged infringer has lawfully discovered the secret through independent research the owner has no rights as against that person.[31] The owner is protected, however, from an infringer who obtains his secret by unlawful means even if the secret had entered the public domain.[32] Thus, the disclosure of a secret through theft[33] or breach of contract[34] in no way infringes upon the owner's rights.

Inherent in the secrecy requirement of trade secret protection are two undesirable features. Even though his rights are not forfeited in each instance of secrecy loss, it is this possibility which is of primary concern when the software owner is selecting a protective device. Once the trade secret rights in soft-

---

is "general" while object code is "specifically adapted" to the particular computer using the program. An example of the object code form of the source code listed in note 25, *supra*, is:

```
58  10  C  ODO
18  21
D2  02  2  021  C  059
58  FO  1  030
05  EF
50  10  D  1F8
58  80  D  1F8

58  50  C  024
07  F5

58  10  C  04C
07  F1
D7  D3  E  6D40309
```

27. Telex Corp. v. IBM, 367 F. Supp. 258, 325 (N.D. Okla. 1973).

28. The fact that the object code form of the program was in daily use at IBM did not lessen the program's "secrecy". Object code can be utilized only by the computer which generated it. Furthermore, as can be seen by the example of object code *supra* note 26, object code is unintelligible to the lay person. Although it is theoretically possible for an expert programmer to reproduce source code from object code, the task is so immense in a complex program as to be fruitless.

29. *Cf.* Telex Corp. v. IBM, 367 F. Supp. 258 (N.D. Okla. 1973); Com-Share v. Computer Complex, Inc., 338 F. Supp. 1229, 1234 (E.D. Mich. 1971).

30. Electronic Data Systems Corp. v. Kinder, 360 F. Supp. 1044 (N.D. Tex. 1973).

31. Drew Chemical Corp. v. Star Chemical Co., 258 F. Supp. 827 (W.D. Mo. 1966).

32. Smith v. Dravo, 203 F.2d 369 (7th Cir. 1953).

33. Telex Corp. v. IBM, 367 F. Supp. 258 (N.D. Okla. 1973).

34. Com-Share, Inc. v. Computer Complex, Inc., 338 F. Supp. 1229 (E.D. Mich. 1971).

ware are lost, they are lost forever and the "owner" then has no greater rights in the software than does the general public.[35] This is the most significant disadvantage in trade secret protection from the standpoint of the software owner. From the perspective of the general public, the major disadvantage to the utilization of the trade secret as a protective device is that they derive no direct benefit from the protected property because of their lack of access thereto.

The problem lies in the lack of compromise between the competing interests at stake; if secrecy is maintained, the public interest of disclosure is not served; if secrecy is lost, the trade secret owner has no rights in the property.

Although trade secret law is the most obvious and natural method of protecting software, other systems of protection must be considered because of the disadvantages inherent in secrecy as a protective device. Both the copyright system and the patent system overcome the major disadvantages of trade secret protection by providing a compromise between the competing interests of the public and the software owner. That is, under these systems, in exchange for a limited monopoly on the software, its owner makes a full disclosure of the property to the public. Thus, unlike the protection provided by the trade secret, these devices grant the public access to the concerned property as well as provide the owner with an adequate degree of protection in the form of remuneration.

II. Patent Protection

Although trade secret law is the most widely-used method of protection, it is patent law which provides the most effective protection.[36] Patent law grants the owner of an invention a monopoly over the "use, manufacture and sale"[37] of his invention for seventeen years.[38] The scope of protection is much wider than that granted by copyright law, which protects only against copying.[39] Although patent law does not require that the secrecy surrounding the property be maintained and does permit public use of the software "invention," there are countervailing considerations which have caused hesitancy among software owners to use the patent system. Of primary concern is the United States Su-

35. *See* University Computing Co. v. Lykes-Youngstown Corp., 504 F.2d 518, 535 (5th Cir. 1974); Electronic Data Systems Corp. v. Kinder, 360 F. Supp. 1044 (N.D. Tex. 1973); CALLMANN, *supra* note 7, § 53.3.

36. In general the patent system consists of a trade between the public and the owner of the invention. In exchange for a limited monopoly granted to him by the government, the owner of an invention discloses it so that the advance in knowledge represented by the invention is a benefit to the general public. For a more thorough discussion of the United States system of patent, see 1 WALKER ON PATENTS, ch. I (2d Deller's ed. 1964). A good historical discussion of the patent system is contained in Comment, *Computer Program Classification: A Limitation on Program Patentability as a Process*, 53 ORE. L. REV. 501, 504-10 (1974) [hereinafter cited as *Computer Program Classification*].

37. *See* Zenith Radio Corp. v. Hazeltine Research, Inc., 395 U.S. 100 (1969); 35 U.S.C. §§ 101, 271 (1970).

38. 35 U.S.C. § 154 (1970).

39. *See* 17 U.S.C. § 1 (1970).

preme Court's rejection of software patents both times the question of software patentability has been presented to it.[40]

Another drawback to the use of patent protection for software lies in the limited range of subject matter which may be patented. Only processes and methods, machines, articles of manufacture, composition of matter, designs and new plants are proper subject matter of patent.[41] Therefore, the inventor must demonstrate that his invention falls into one of these statutory classes and his failure to do so will result in the rejection of his application.[42]

A. *Treatment of Software Applications by the Court of Customs and Patent Appeals*

1. *Apparatus Claims*[43]

The earliest reported decision in which the question of software patentability was at issue was *Ex parte King*.[44] In *King*, the combination of a computer program and the computer in which it was stored during operation was held patentable as a machine. This approach to subject matter classification of software has been approved by the Court of Customs and Patent Appeals (CCPA) in several cases since *King*.[45] In *In re Prater*,[46] the court, relying on *King*, validated patent claims for a computer program in conjunction with the machine in which it was stored. The rationale of both *King* and *Prater* was perhaps most appropriately summarized in *In re Bernhart*[47] wherein the court observed:

> [I]f a machine is programmed in a certain new and unobvious way, it is physically different from the machine without that program; its memory elements are differently arranged. The fact that these physical changes are invisible to the eye should not tempt us to conclude that the machine has not been changed. If a new machine has not been invented, certainly a "new and useful improvement" of the unprogrammed machine has been. . . .[48]

A patent obtained through claiming software as an "apparatus" does not, however, provide complete protection. This type of patent protects the program as a component of a machine but does not extend to it any individual protec-

---

40. Dann v. Johnston, 425 U.S. 219 (1976); Gottschalk v. Benson, 409 U.S. 63 (1972).
41. 35 U.S.C. § 101 (1970).
42. *See, e.g., In re* Prater, 415 F.2d 1393 (C.C.P.A. 1969).
43. The term "apparatus claim" is used to designate a claim in a patent application for software in which the software and the computer in which it is embodied are claimed as a single machine. *See, e.g., In re* Prater, 415 F.2d 1393 (C.C.P.A. 1969). A further discussion of apparatus claims is contained in *Computer Program Classification, supra* note 36, at 512.
44. 146 U.S.P.Q. 590 (Pat. Off. Bd. App. 1964).
45. *See, e.g., In re* Bernhart, 417 F.2d 1395 (C.C.P.A. 1969); *In re* Prater, 415 F.2d 1393 (C.C.P.A. 1969).
46. 415 F.2d 1393 (C.C.P.A. 1969). This cite is to the rehearing decision by the CCPA in the Prater application. Courts and commentators alike refer to this second decision as *Prater II*; it will be so referenced in this Note.
47. 417 F.2d 1395 (C.C.P.A. 1969).
48. *In re* Bernhart, 417 F.2d 1395, 1400 (C.C.P.A. 1969).

tion. An additional potential limitation on the use of apparatus claims may be a growing reluctance on the part of the CCPA to validate them. In what could well be the future attitude of the court, the dissent to *In re Johnston*[49] labeled as a "legal fiction" the theory that a new program in an existing computer constituted a "new machine."[50]

Nonetheless, patents continue to be sought and granted for programs classified as machines.[51] The consistency with which apparatus claims for software have been validated[52] may be due to a feeling by the CCPA that no other statutory subject matter class is *more* appropriate for software.[53]

2. *Method Claims*[54]

The alternate statutory subject matter class which has been found applicable to software is that of a process.[55] A statutory process is a "process, art or method, and includes a new use of a known process, machine, manufacture, composition of matter, or material."[56] The rationale for use of this classification is that the computer is an "old" machine and that operation of the computer with a new program is a new use of an old machine, and hence a statutory process.[57]

Although it is well settled that a process can be a patentable invention,[58] attempts to place software in the process category have met with uneven success in the CCPA.[59] This is due in large part to the gradual demise in the CCPA of the so-called "mental steps" doctrine.[60] Under this doctrine, an application for a patent on software may be rejected under section 101[61] as "reading on"[62]

---

49. 502 F.2d 765 (C.C.P.A. 1974), *rev'd sub nom. on other grounds*, Dann v. Johnston, 425 U.S. 219 (1976).
50. *In re* Johnston, 502 F.2d 765, 773 (C.C.P.A. 1974) (Rich, J., dissenting).
51. *Id.*; *In re* Knowlton, 481 F.2d 1357 (C.C.P.A. 1973); *In re* McIlroy, 442 F.2d 1397 (C.C.P.A. 1971).
52. No reported cases were discovered in which a claim drawn in terms of program plus computer was rejected as non-statutory under 35 U.S.C. § 101 (1970).
53. *In re* Johnston, 502 F.2d 765, 773 (C.C.P.A. 1974) (Rich, J., dissenting).
54. A "method claim" is a patent application claim in which the software is claimed as a process.
55. 35 U.S.C. § 101 (1970).
56. 35 U.S.C. § 100(b) (1970).
57. *See, e.g., In re* Mahony, 421 F.2d 742 (C.C.P.A. 1970).
58. Cochrane v. Deener, 94 U.S. 780 (1877).
59. *Compare In re* Prater, 415 F.2d 1393 (C.C.P.A. 1969) (method claims disallowed) *with In re* Musgrave, 431 F.2d 882 (C.C.P.A. 1970) (method claims allowed).
60. *See, e.g., In re* Abrams, 188 F.2d 165 (C.C.P.A. 1951) for an early discussion of this doctrine.
61. 35 U.S.C. § 101 (1970).
62. "Reading on" is a shorthand phrase used in patent practice to signify that a broadly drafted claim has described a device or process other than the invention of that particular patent application. For example, if the invention in question is a four-legged stool and it is described in the claims as "a horizontal planar rectangular device, a leg attached to each corner thereof", the application might be rejected as "reading on" the invention of a table. The claim does describe a simple stool, but is so broadly drawn that it also describes a table. The underlying defect in this claim is that the invention cannot be discerned with particularity.

non-statutory subject matter.[63] That is, if the method claim is written in language broad enough to describe a process which can be performed entirely mentally,[64] or performed mentally with the assistance of pencil and paper,[65] then the claim must be rejected. The rationale for rejection is prevention of a monopoly on a process which is entirely cognitive.

The mental steps doctrine has evolved from *In re Abrams*,[66] where it was established in the form of three rules of law,[67] to *In re Musgrave*,[68] in which the doctrine was apparently eradicated.[69] At a minimum, the broad test adopted in *Musgrave* has reduced the effectiveness of a "mental steps" rejection of a method claim for software.[70] Under that test, "[a]ll that is necessary . . . to make a sequence of operational steps a statutory 'process' within 35 U.S.C. § 101 is that it be in the technological arts so as to be in consonance with the Constitutional purpose to promote the progress of 'useful arts.' "[71] In adopting this test the court stated that method claims are not non-statutory "merely because some or all the steps therein can also be carried out in or with the aid of the human mind or because it may be necessary for one performing the processes to think."[72]

This test is currently applied to method claims which read the CCPA.[73] These applications of the *Musgrave* test have clarified the position of the CCPA with regard to the mental steps rejection. The result of these decisions is that in the CCPA "very little remains of the 'mental steps' doctrine."[74] Although

---

63. *See, e.g., In re* Prater, 415 F.2d 1393 (C.C.P.A. 1969).
64. *In re* Abrams, 188 F.2d 165, 166 (C.C.P.A. 1951).
65. *In re* Prater, 415 F.2d 1393 (C.C.P.A. 1969).
66. 188 F.2d 165 (C.C.P.A. 1951).
67. The rules adopted by the court were suggested by appellant in his brief as:

> 1. If all the steps of a method claim are purely mental in character, the subject matter thereof is not patentable within the meaning of the patent statutes.
>
> 2. If a method claim embodies both positive and physical steps as well as so-called mental steps, yet the alleged novelty or advance over the art resides in one or more of the so-called mental steps, then the claim is considered unpatentable for the same reason that it would be if all the steps were purely mental in character.
>
> 3. If a method claim embodies both positive and physical steps as well as so-called mental steps, yet the novelty or advance over the art resides in one or more of the positive and physical steps and the so-called mental step or steps are incidental parts of the process which are essential to define, qualify or limit its scope, then the claim is patentable and not subject to the objection contained in 1 and 2 above.

In re Abrams, 188 F.2d 165, 166 (C.C.P.A. 1951).
68. 431 F.2d 882 (C.C.P.A. 1970).
69. This view was taken by Judge Baldwin, concurring in *Musgrave*, wherein he stated that "very little remains of the 'mental steps' doctrine." *In re* Musgrave, 431 F.2d 882, 895 (C.C.P.A. 1970).
70. *See, e.g., In re* Benson, 441 F.2d 682 (C.C.P.A. 1971), *rev'd sub nom. without reaching this issue*, Gottschalk v. Benson, 409 U.S. 63 (1972); *In re* Waldbaum, 457 F.2d 997 (C.C.P.A. 1972).
71. *In re* Musgrave, 431 F.2d 882, 893 (C.C.P.A. 1970).
72. *Id.*
73. The test adopted by the *Musgrave* court was explicitly followed by the Court of Customs and Patent Appeals in *In re Waldbaum*, 457 F.2d 997 (C.C.P.A. 1972) and *In re Benson*, 441 F.2d 682 (C.C.P.A. 1971).
74. *In re* Benson, 441 F.2d 682, 687 (C.C.P.A. 1971), *citing with approval In re* Musgrave, 431 F.2d 882 (C.C.P.A. 1970).

the status of the mental steps doctrine in the CCPA is now clear, the attitude of the United States Supreme Court toward that doctrine has not yet been articulated.[75] Thus, software owners are uncertain as to the likelihood of success of method claims on their software.[76] This uncertainty is evidenced by a complete lack of cases discussing the mental steps doctrine since *In re Benson*.[77]

Absence of discussion of the mental steps doctrine in recent cases may simply be due to a lack of rejections of software claims on this ground by the patent office. More recent cases have all come to the CCPA on appeal of a rejection based on 35 U.S.C.A. section 112 (1975).[78] This section requires that the specification portion of a patent application disclose the invention to the extent sufficient to "enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same. . . ."[79] Disclosure insures that the public acquires access to the invention in exchange for the limited monopoly granted to the inventor.

The patent office has taken two different approaches in rejecting software applications for non-disclosure. One type of section 112 rejection can be seen in *Prater II*.[80] This rejection is very similar to a "mental steps" rejection under section 101. The cause of rejection in each case is the same, but the rationale for rejection is different. If the claim is rejected under section 101 for "reading on" mental steps, the rationale for rejection is that non-statutory subject matter is being claimed. If the claim is rejected under section 112 for non-disclosure the reasoning is that one skilled in the art would be unable to discern the invention; he would be unable to tell if the invention was the computer program or the mental steps. A section 112 rejection of this type was used by the patent office only in conjunction with a section 101 "mental steps" rejection.[81] As a result of the decline of the "mental steps" doctrine, this section 112 basis of rejection has fallen into disuse.

The more traditional approach is illustrated in *In re Comstock*,[82] which involved an apparatus claim that had been rejected by the patent office for not

---

75. The issue was left undiscussed both times it was presented to the Court. *See* Dann v. Johnston, 425 U.S. 219 (1976); Gottschalk v. Benson, 409 U.S. 63 (1972).

76. Oler, *Statutory Copyright Protection for Electronic Digital Computer Programs: Administrative Considerations*, 7 L. & COMPUTER TECH. 96, 100 (1974) [hereinafter cited as Oler].

77. 441 F.2d 682 (C.C.P.A. 1971); *cf.* Comment, *Patentability: Piecing Together the Computer Software Patent Puzzle*, 19 ST. LOUIS U.L.J. 351 (1975) [hereinafter cited as *Patentability*]. The author of this exhaustive chronological survey of cases also fails to locate a case since *Waldbaum* discussing the "mental steps" issue.

78. *In re* Johnston, 502 F.2d 765 (C.C.P.A. 1974), *rev'd sub nom.*, Dann v. Johnston, 425 U.S. 219 (1976); *In re* Knowlton, 500 F.2d 566 (C.C.P.A. 1974); *In re* Scarbrough, 500 F.2d 560 (C.C.P.A. 1974); *In re* Brandstadter, 484 F.2d 1395 (C.C.P.A. 1973); *In re* Doyle, 482 F.2d 1385 (C.C.P.A. 1973), *cert. denied*, 416 U.S. 935 (1974); *In re* Knowlton, 481 F.2d 1357 (C.C.P.A. 1973); *In re* Comstock, 481 F.2d 905 (C.C.P.A. 1973).

79. 35 U.S.C. § 112 (1970).

80. *In re* Prater, 415 F.2d 1393 (C.C.P.A. 1969).

81. *See, e.g., In re* Foster, 438 F.2d 1011 (C.C.P.A. 1971); *In re* Mahony, 421 F.2d 742 (C.C.P.A. 1970); *In re* Prater, 415 F.2d 1393 (C.C.P.A. 1969).

82. 481 F.2d 905 (C.C.P.A. 1973).

disclosing a general purpose computer apparatus which would comply with the claims. The CCPA reversed this rejection, stating that disclosure was sufficient if it showed the program as embodied within the computer, and that it was not necessary to describe the computer in minute detail.[83] In *Comstock*, the general theory of rejection was that the invention as disclosed in the application could not be made and used by the skilled artisan. As can be seen from the discussion of these cases, the basis of this rejection is lack of sufficient information to possess the invention.

In *In re Ghiron*,[84] where the invention was a "method of facilitating transfers dictated by programs for processing data in a computer operating in overlap mode,"[85] the court affirmed the patent office's section 112 rejection. In *Ghiron*, the court held that a block diagram and flow chart of the program did not constitute sufficient disclosure to make possible use of the invention. In contrast, an application consisting of a block diagram of the program, a program listing and a narrative description of each instruction was held to be sufficient disclosure to avoid a section 112 rejection in *In re Knowlton*.[86] The application in *Knowlton* seems to exemplify the minimum amount of disclosure of a computer program which will be regarded as sufficient. Less disclosure has always resulted in a section 112 rejection.[87] The disclosure is also insufficient if practice of a disclosed invention necessitates an unreasonable amount of additional work and experimentation.[88] A reading of the first *Knowlton*[89] case with *In re Doyle*[90] and *In re Brandstadter*[91] indicates that, at a minimum, a program listing is required to avoid rejection for non-disclosure.

B. *Treatment of Software Applications by the United States Supreme Court*

The United States Supreme Court has had two opportunities to assist the

83. *In re* Comstock, 481 F.2d 905, 909 (C.C.P.A. 1973).
84. 442 F.2d 985 (C.C.P.A. 1971).
85. *In re* Ghiron, 442 F.2d 985, 986 (C.C.P.A. 1971). *See also In re* Gunn, 537 F.2d 1123 (C.C.P.A. 1976).
86. 481 F.2d 1357 (C.C.P.A. 1973).
87. *In re* Gunn, 537 F.2d 1123 (C.C.P.A. 1976) (block diagram of devices disclosing only their generic names was insufficient disclosure under section 112 where the invention was the apparatus and computerized method of computing the amount of postage and printing it on the mail); *In re* Doyle, 482 F.2d 1385 (C.C.P.A. 1973), *cert. denied*, 416 U.S. 935 (1974) (failure to disclose in any form the computer programs necessary to implement a computerized method was justification for a section 112 rejection); *In re* Brandstadter, 484 F.2d 1395 (C.C.P.A. 1973) (again, failure to disclose the computer program); *In re* Scarbrough, 500 F.2d 560 (C.C.P.A. 1974) (disclosure limited to the generic names of devices to be used in the process is insufficient disclosure). It is interesting that one inventor who successfully bypassed a section 112 rejection in one case was rejected in his second application for a patent on software. *Compare In re* Knowlton, 481 F.2d 1357 (C.C.P.A. 1973) (a program to process "list information" such as inventories, personnel files, business accounts, etc.) *with In re* Knowlton, 500 F.2d 566 (C.C.P.A. 1974) (a system and method for detecting programming errors in programs for electronic data processors or compilers).
88. *In re* Scarbrough, 500 F.2d 560 (C.C.P.A. 1974).
89. *In re* Knowlton, 481 F.2d 1357 (C.C.P.A. 1973).
90. 482 F.2d 1385 (C.C.P.A. 1973), *cert. denied*, 416 U.S. 935 (1974).
91. 484 F.2d 1395 (C.C.P.A. 1973).

CCPA in its struggle with applications for patent protection of computer software.[92] Rather than discussing issues which have concerned the CCPA, the Court in both cases based its rulings on established patent law principles.

In *Gottschalk v. Benson*,[93] a patent on software was invalidated because its effect was to give the inventor a monopoly on a mathematical formula.[94] The principle that a mathematical formula may not be patented has long been settled,[95] so the Supreme Court in *Benson* broke no new ground with its ruling. The *Benson* invention was a computerized formula to convert binary coded decimal numbers to pure binary.[96] Method claims on this formula were rejected by the patent office under section 101 as containing "mental steps" and hence covering non-statutory subject matter.[97] The CCPA, relying on the *Musgrave*[98] decision,[99] reversed and held that the process was statutory.

Even though the "mental steps" issue was squarely presented to the Court, it was not discussed. Other questions of importance to the software community were discussed, but, unfortunately, were left unanswered. No decision was reached on whether a computer program was patentable.[100] The Court also expressly left unanswered the question of whether a program which was operable with several apparatuses could be patentable.[101] Since the question of program patentability was explicitly left open, it would appear that until Congress addresses this matter, programs which satisfy the statutory requirements may be patented.

The Court's broad hint to Congress that legislation concerning software patentability is needed[102] may be the only positive aspect of the *Benson* decision. Since the question of whether software is patentable as a process was not answered by the Court, the door to future applications for patents on software inventions has been left open. Furthermore, having refused to address the "mental steps" doctrine, the Court has not provided the patent office and the CCPA with guidance on the crucial issue concerning software patentability.[103]

It had been hoped[104] that the confusion remaining after *Benson* would be resolved by the Court when it decided *Dann v. Johnson*.[105] Unfortunately, that case added nothing to the law of software patentability.

---

92. Dann v. Johnston, 425 U.S. 219 (1976); Gottschalk v. Benson, 409 U.S. 63 (1972).
93. 409 U.S. 63 (1972).
94. Gottschalk v. Benson, 409 U.S. 63, 67-68 (1972).
95. *See, e.g.*, McKay Radio & Telegraph Co. v. Radio Corp. of America, 306 U.S. 86 (1939). This case was the principal authority relied upon in the *Benson* decision.
96. Gottschalk v. Benson, 409 U.S. 63 (1972).
97. *In re* Benson, 441 F.2d 682, 684 (C.C.P.A. 1971).
98. *In re* Musgrave, 431 F.2d 882 (C.C.P.A. 1970).
99. *In re* Benson, 441 F.2d 682, 687 (C.C.P.A. 1971).
100. Gottschalk v. Benson, 409 U.S. 63, 71 (1972).
101. *Id.*
102. *Id.* at 73.
103. This lack of guidance is certainly a major factor in the current dormancy of the "mental steps" doctrine in the CCPA.
104. *Patentability, supra* note 77, at 374.
105. 96 S. Ct. 1393 (1976), *rev'g In re* Johnston, 502 F.2d 765 (C.C.P.A. 1974).

The invention in *Johnston* was a " 'machine system for automatic record-keeping of bank checks and deposits.' "[106] The method claims in the patent application on this invention were rejected by the patent office on several grounds: 1) for not distinctly claiming the invention as required by section 112; 2) for claiming subject matter not falling into one of the section 101 classes;[107] (the board of appeals held that the invention related to a banking function and was an advance of the social arts, not the technological arts, thereby effectively skirting the *Musgrave* test); and 3) for claiming an invention which was obvious and therefore disallowed under section 103.[108]

The CCPA disagreed with each basis for the rejection and reversed. The section 101 holding was reversed on the facts. The court thought that the invention was a record keeping *machine* system and therefore clearly technological.[109] The grounds for rejection under sections 103 and 112 were summarily reversed.[110]

The Supreme Court found the issue of obviousness to be dispositive and reversed the CCPA. Relying on *Graham v. John Deere Co.*,[111] the leading case on the scope of section 103, the Court found the invention to be obvious.[112] Applying the traditional standards of obviousness found in the *Graham* case, the Court held that Johnston's invention was not sufficiently different from prior systems of bank record keeping " 'to render the claimed subject matter unobvious to one skilled in the applicable art . . . .' "[113] Thus, although all unresolved issues of software patentability were thoroughly briefed, the Supreme Court again left them undecided.

This decision is likely to have an effect similar to that of *Gottschalk v. Benson*[114]—uncertainty. However, it is conceivable that the effect will be quite beneficial to the law of software patentability. Both *Gottschalk* and *Johnston* may well be read as a clear indication by the Supreme Court that it will not answer this question. If so, the decisions may encourage Congress to legislate in the area of software patentability. The immediate effect, however, will be to inhibit software owners in seeking patent protection for software.

The confusion caused by *Benson* and the failure of the Court to resolve it in *Johnston*, has made owners of programs unwilling to entrust their valuable property to a system of protection which rests on so tenuous a foundation.[115]

---

106. Dann v. Johnston, 425 U.S. 219, 220 (1976).
107. The board of appeals held that the invention related to a banking function and was an advance of the social arts, not the technological arts, thereby effectively skirting the *Musgrave* test.
108. 35 U.S.C. § 103 (1970).
109. *In re* Johnston, 502 F.2d 765, 770-71 (C.C.P.A. 1974).
110. *Id.* at 770-72.
111. 383 U.S. 1 (1966).
112. Dann v. Johnston, 425 U.S. 219, 229 (1976).
113. *Id.* at 228, *quoting In re* Johnston, 502 F.2d 765, 772 (C.C.P.A. 1974) (Markey, C.J., dissenting).
114. 409 U.S. 63 (1972).
115. *See* Note, *Computer Software—Beyond the Limits of Existing Propietary Protection Policy*, 40 BROOKLYN L. REV. 116, 141 (1974).

As a result, interest in obtaining copyright protection for software has recently increased.[116]

III. Copyright Protection

Although *Benson* and *Johnston* have intensified interest in the use of copyright law to protect software, there has been interest in this form of protection for nearly fifteen years.[117] The possibility of registering computer programs for copyright protection was officially recognized by the Copyright Office in an announcement on May 19, 1964.[118] This announcement provides a general description of computer programs as "either a set of operating instructions for a computer or a compilation of reference information to be drawn upon by the computer in solving problems."[119] The announcement states that the two basic questions in regard to registrability are:

> (1) Whether a program as such is the "writing of an author" and thus copyrightable, and (2) whether a reproduction of the program in a form actually used to operate or be "read" by a machine is a "copy" that can be accepted for copyright registration.[120]

The Copyright Office believed that resolution of these two questions in favor of registrability was doubtful. Nevertheless, the Office followed its policy of resolving doubtful issues in favor of registration wherever possible and decided to accept registration of computer programs as books.[121] The announcement stated that:

> Registration for a computer program will be considered if:
>
> The elements of assembling, selecting, arranging, editing, and literary expression that went into the compilation of the program are sufficient to constitute original authorship.
>
> The program has been published with the required copyright notice; that is, "copies" (i.e., reproductions of the program in a form perceptible to the human eye) bearing the notice have been distributed or made available to the public. Two copies deposited for registration consist of two complete copies of the program in the form as first published. If the first publication was in a form (such as machine-readable tape) that cannot be perceived visually or read by humans, a visually perceptible reproduction or description (such as a print-out of the program) must also be deposited.[122]

---

116. Oler, *supra* note 76.
117. "So far as the [Copyright] Office is aware . . . the first deposit for copyright registration of a computer program in the form of a tape was effected by North American Aviation, Inc., on November 30, 1961." Cary, *Copyright Registration and Computer Programs*, 11 Bull. Copyright Soc'y 362, 363 (1964) [hereinafter cited as Cary].
118. *Copyright Registration for Computer Programs—Announcement From the Copyright Office*, 11 Bull. Copyright Soc'y 361 (1964) [hereinafter cited as *Announcement*].
119. *Id.*
120. *Id.*
121. *Id.*
122. *Id.*

This announcement was no real surprise to those familiar with the Copyright Office since it merely reaffirmed well-established copyright principles.[123] Protection has long been granted to subject matter that is at least arguably similar to computer programs. In *Reiss v. National Quotation Bureau, Inc.*,[124] Judge Learned Hand, writing for the court, held that a group of coined words[125] without any basic meaning could be copyrighted.[126] In light of *Reiss*, it could be argued that a computer program, even though it may appear nonsensical, is the "writing of an author." Actually, however, this element of copyrightability is more easily met by a computer program than by the words in *Reiss* since a computer program has meaning and can be readily understood by one familiar with the programming language used.[127]

Often computer programs take the form of a deck of punched cards. It may be suggested that this form is not a copy of the program suitable for registration since it cannot be read.[128] This argument is untenable because the cards *can* be read by one familiar with the code upon which the punches are based.[129] Furthermore, it is customary for cards to be "interpreted;" that is, the holes are translated to English which is printed at the top of the card, thereby enabling a copyright examiner to read the program in that form.

A subject similar to punched cards was discussed in *White-Smith Publishing Co. v. Appolo Co.*[130] wherein the Court held that a perforated player piano roll was not a "copy."[131] Because a player-piano roll is very much like a deck of punched cards, *White-Smith* would seem to control with the result being that a card deck is not a "copy." The significance of *White-Smith* lies in its rationale: a piano roll was held not to be a "copy" because it could not be read.[132] Since

123. Cary, *supra* note 117.
124. 276 F. 717 (S.D.N.Y. 1921).
125. These were simply five-letter words made up of randomly selected letters, all capable of being pronounced, but having no meaning. Reiss v. National Quotation Bureau, Inc., 276 F.2d 717, 718 (S.D.N.Y. 1921).
126. *Id.*
127. There are many different languages and many types of languages. For example, Basic Assembler Language (BAL) and COBOL are generally used for commercial purposes because they are word-handling languages. They are capable of efficiently performing record keeping functions. FORTRAN and ALGOL are languages which are well adapted to scientific functions and are used to solve problems requiring higher mathematics.

BAL appears to be gibberish to the uninitiated, as well as to most programmers, but the so-called "higher level" languages COBOL and FORTRAN read much like a set of instructions in English (see the coding at note 25 *supra* for an example of COBOL coding, and the coding at note 26 *supra* for an example of BAL coding). One who is totally unfamiliar with the mechanical operation of a computer can, nonetheless, instruct the computer using these languages. The person who has not previously used these languages would be able to look at a program written in COBOL or FORTRAN and at least understand that it had meaning; he would probably have a glimmer of understanding of the function of the program as well.
128. *Cf.* Lawlor, *Copyright Aspects of Computer Usage*, 11 BULL. COPYRIGHT SOC'Y 380, 402 (1964) (not a copy so as to constitute an infringement).
129. From experience the author can assure the reader that this is so. A proficient programmer can quite amaze the uninitiated with his ability to "read holes". This is necessary in his job because occasionally cards are not interpreted. *See* text following note 129 *infra* for an explanation of "interpreted."
130. 209 U.S. 1 (1908).
131. White-Smith Publishing Co. v. Appolo Co., 209 U.S. 1, 18 (1908).
132. *Id.*

a program can be read, however, *White-Smith* would appear to be no bar to finding that a computer deck is a copy.

Because of past protection of subject matter similar to computer programs as "writings of an author," because computer programs may be written in a form suitable for registration, and because of its policy of resolving doubtful issues in favor of registrability, the Copyright Office decided to register computer programs which comply with published regulations.[133] This decision remains the policy of the Copyright Office[134] and thus, copyright protection for software *is* available.

The discussion of the Copyright Office's policy concerning copyrightability of software indicates a few of the problems encountered with this form of protection. The policy of allowing registration resolves some, but not all, difficulties of copyright protection for computer programs. A policy of registration resolves only the issue of whether software may be registered; it does not alleviate the problem regarding the limited scope of protection which is inherent in the copyright law.

The major disadvantage of this form of protection is the nature of the rights granted.[135] The general rule is that copyright protects the *writing* from copying by others, but does not protect against another's use of the *idea* contained therein.[136] Thus the protection granted is limited and incomplete, quite unlike protection granted by patent. A patent is a grant to its owner of a monopoly on use of the invention's embodiment of the idea.[137] This distinction was well expressed by the court in *Baker v. Selden*,[138] wherein Justice Bradley stated that:

> The description of the art in a book, though entitled to the benefit of copyright, lays no foundation for an exclusive claim to the art itself. The object of the one is explanation; the object of the other is use. The former may be secured by copyright. The latter can only be secured, if it can be secured at all, by letters-patent.[139]

The significance of this difference is that the copyrighted program is not protected against the development of an identical computer program, so long as the latter is independently developed.[140] The extent of the protection offered by the copyright is, however, consistent with the general nature of the material

---

133. *Announcement, supra* note 118.

134. Copyright Office, Circular 61, *Computer Programs*, Feb., 1974 [hereinafter cited as Circular].

135. *See* 17 U.S.C. § 1 (1970).

136. Baker v. Selden, 101 U.S. 99 (1879); Roth Greeting Cards v. United Card Co., 429 F.2d 1106 (9th Cir. 1970); McGraw-Hill, Inc. v. Worth Publishers, Inc., 335 F. Supp. 415 (S.D.N.Y. 1971); Norman v. CBS, Inc., 333 F. Supp. 788 (S.D.N.Y. 1971); Gardenia Flowers, Inc. v. Joseph Markovits, Inc., 280 F. Supp. 776 (S.D.N.Y. 1968).

137. Baker v. Selden, 101 U.S. 99 (1879).

138. 101 U.S. 99 (1879).

139. Baker v. Selden, 101 U.S. 99, 105 (1879). Of course, not even a patent on the program would entitle its owner to a monopoly on the naked idea. Gottschalk v. Benson, 409 U.S. 63 (1972); MacKay Radio & Telegraph Co. v. Radio Corp. of America, 306 U.S. 86 (1939).

140. *Cf.* 17 U.S.C. § 1(f) (1970). For a general discussion see Note, *Protection of Computer Programs: Resurrection of the Standard*, 50 NOTRE DAME LAW. 333 (1975).

concerned.[141] It is unlikely that most books, musical works and works of art will be identically, yet independently, created. Thus, providing protection against that possibility is unnecessary in the usual copyright, for there is only a small gap in the protection granted. This gap takes on additional significance, however, when the nature of a computer program is considered. A computer program is often a composite of ideas developed and tested in previous programs. A program may be merely the next step in a line of development, merely an extension of "prior art."[142] Thus it is not unlikely that two wholly independent programs might develop in an identical manner. In such an instance, neither is afforded protection from infringement by the other.[143]

Perhaps it is defensible for a good faith, independent development to not be regarded as an infringement. The independent developer has used his own resources and creativity and no misappropriation is involved. The difficulty lies in the fact that a computer program is more like an invention than a work of art. Because of this characteristic, software should be protected against unauthorized use and not merely against unauthorized copying. The protection currently granted by the copyright statute does not provide the needed protection. It is inadequate to safeguard the entire computer program, the writing and the use of the underlying idea.

Two final difficulties in the use of copyright protection for computer programs should also be noted. "Publication" of the program is required to secure rights under the statute.[144] In addition, the copy published must have a notice of copyright affixed to it.[145] The difficulties in meeting these requirements have, for the most part, been overcome, but in the early days of computer usage they were thought to present serious obstacles to registration. Early programs often took the form of a card deck or a paper or magnetic tape. Since the purpose of publishing copyrighted works is disclosure, it was not clear whether programs in these forms could be "published" since they were not intelligible to the general public. There was also a problem in affixing notice to a magnetic tape. These difficulties have been surmounted by technological advances. Programs, in whatever form they are used by the computer, may now be readily translated to a printed form, publication being thereby made meaningful. Furthermore, notice of copyright can be made internal to a program so that whenever the program is printed, the printout contains the requisite notice. Publication has been made even more meaningful today because of a fifth registration requirement applicable to computer programs. Currently an explanation of the "way in which the program was made available to the public, and [of] the form in which the copies were published is required before registration is ap-

141. Sheldon v. Metro-Goldwyn Pictures Corp., 81 F.2d 49 (2d Cir. 1936).
142. *Cf.* 35 U.S.C. §§ 102, 103 (1970). This, of course, was the reason given by the Supreme Court in its rejection of the software patent application. Dann v. Johnston, 96 S. Ct. 1393 (1976).
143. Sheldon v. Metro-Goldwyn Pictures Corp., 81 F.2d 49 (2d Cir. 1936).
144. 17 U.S.C. § 10 (1970).
145. *Id.*

proved."[146] Thus, if the form itself renders the disclosure to the public ineffective, the required explanation will cure it.

Generally, copyright protection of computer programs is unsatisfactory in that it affords no protection against another's use of the program's embodiment of its underlying idea. There is no immediate prospect for increasing the scope of copyright protection. Although the copyright law, Title 17 of the United States Code, is undergoing extensive revision,[147] protection of computer programs by the new law remains limited. Rather than broadening the scope of protection, the new legislation has reinforced the current limitations.[148] Section 102, the subject matter section of revised Title 17, provides: "(b) In 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, regardless of the form in which it is described, explained, illustrated, or embodied in such work."[149] The Register of Copyrights regards this section as merely a clarification of existing copyright law.[150] The intent of the new section is "not to enlarge or contract the scope of protection under the present law, but rather 'to restate, in the context of the new single Federal system of copyright, that the basic dichotomy between expression and idea remains unchanged.' "[151] Specifically, it is intended to "disclaim any intention to protect a programmer's algorithms under the bill."[152] One encouraging aspect of the new legislation, however, is that even though the scope of protection remains limited, computer programs are still an intended subject matter of copyright protection.[153]

IV. Proposal

It is apparent that there exists no method of effectively protecting the complete computer program. Trade secret protection is effective—but only so long as secrecy can be maintained. Patent protection may again become popular since it provides broad protection, but until the questions left by *Gottschalk v. Benson*[154] are resolved, patents will continue to be regarded as too uncertain to be effective. Copyright protection is ineffective because the scope of protection is too narrow in that it does not encompass the inventive aspects of software. The obvious solution is a system of protection designed especially for software.

Users of software might naturally be expected to be the first to recognize the inadequacies of any current system of software protection. As early as

146. Circular, *supra* note 134.
147. *See* Copyrights Act, Pub. L. No. 94-553, 90 Stat. 2541 (1976) (to be codified in 17 U.S.C. § 101 *et seq.*).
148. *Id.*, § 101, 90 Stat. 2541, 2541 (1976) (to be codified in 17 U.S.C. § 102(b)).
149. *Id.*
150. Register of Copyrights, Second Supplemental Report on the General Revision of the U.S. Copyright Law: 1975 Revision Bill 2 (Oct. 1975).
151. *Id.* (quoting S. Rep. No. 94-473, 93d Cong., 2d Sess. 54).
152. *Id.*
153. S. Rep. No. 94-473, 93d Cong., 2d Sess. 50, 51.
154. 409 U.S. 63 (1972).

1971, a group of software users proposed a study to "concentrate upon the desirable ingredients of *any* system of legal protection of computer programs, rather than follow rigidly the lines of established legal concepts for the protection of other inventions or works."[155] This proposal certainly indicates dissatisfaction with current methods of protection and evidences a desire on the part of those who use programs to find a system of protection specially adapted to software. Dissatisfaction with statutory protection of software was also indicated by Justice Douglas in *Gottschalk v. Benson*.[156] There can be little doubt that the present means for protecting computer software is unsatisfactory and that a new method of protection is required.[157]

Since protection of computer programs was not anticipated by the drafters of the copyright and patent statutes, the inadequacies of their protection are not surprising. To overcome these inadequacies and to provide disclosure to the public, which is lacking when computer programs are treated as trade secrets, some protective system specially designed for software should be provided. It had been hoped that revision of the copyright statute would yield such a scheme. Unfortunately, instead of providing an innovative protective device, the new legislation reinforces the present limitations of copyright protection. This result has been criticized by the Register of Copyrights in her statement to the House Committee considering the Senate version of the copyright revision bill.[158]

Legal commentators have also suggested that a specially designed system of proprietary protection be instituted.[159] One author suggests a detailed scheme;[160] another, a mere outline.[161] Both, however, are of the opinion that what is required is one system that combines the positive aspects of both patent and copyright statutes. This is clearly the course Congress should follow in providing proprietary protection to computer software. The new protection could grant a monopoly on both the writing *and* use of the program's underlying idea. Thus, protection of the "inventive" aspect of software would be similar to present patent protection, while protection of the "writing" would be analogous to current copyright protection. The protection could be granted for a shorter period of time than that granted to either a patent (seventeen years)[162] or a copyright (twenty-eight years).[163] A shorter period of protection would recognize the rapid advance in technique and technology in the computer industry and would provide incentive for further innovations.

155. Resolution of the Geneva Meeting on Computer Software, March 8-12, 1971, *quoted in* 18 BULL. COPYRIGHT SOC'Y 355 (1971) (emphasis added).
156. 409 U.S. 63, 73 (1972).
157. *See* Galbi, *Proposal for New Legislation to Protect Computer Programming*, 17 BULL. COPYRIGHT SOC'Y 280 (1970) [hereinafter cited as Galbi]; Oler, *supra* note 76; *Beyond the Limits*, *supra* note 5.
158. Ringer, *supra* note 156, at 5.
159. Galbi, *supra* note 157; Scafetta, *supra* note 5, at 398.
160. Galbi, *supra* note 157.
161. Scafetta, *supra* note 5.
162. 35 U.S.C. § 154 (1970).
163. 17 U.S.C. § 24 (1970). A copyright in a work created on or after January 1, 1978, has a term equal to the life of the author and fifty years after the author's death. Copyrights Act, Pub. L. No. 94-553, § 101, 90 Stat. 2541, 2572 (1976) (to be codified in 17 U.S.C. § 302).

Another potential feature of a new system of protection which Congress should consider is a "subject matter" classification scheme for software.[164] There could at least be a distinction made between simple programs and complex programs. Copyright protection for a simple program is probably sufficient since there is no real invention or advance in the "useful arts." Programs such as compilers or other systems of software, however, contain many complex and innovative sections of coding, and because of their inventive aspects deserve protection akin to that provided by the patent system.

The present criteria for obtaining patent protection are utility,[165] novelty,[166] and non-obviousness;[167] the criterion for obtaining copyright protection is originality.[168] All four could be criteria for a proposed system of software protection. As in the present patent and copyright systems, full disclosure to the public would be required in exchange for the grant of protection.

V. Conclusion

Because of the disadvantages inherent in use of any of the three proprietary protection methods currently available for software, it is apparent that a new scheme of protection is required. The special nature of computer software and the drawbacks of current protection methods should be major considerations in any new system of protection of computer software. A new system of protection should recognize that a computer program is an invention as well as a writing and should provide a guarantee to the software owner that neither aspect of his property will be infringed. No other design will overcome the limitations of current methods for safeguarding software and provide fully satisfactory protection.

Mark H. Ogden

---

164. Such a scheme was suggested in *Computer Program Classification*, *supra* note 36, at 519. That scheme is perhaps too complex to be workable; the distinction drawn between classes of programs is probably too fine.
165. 35 U.S.C. § 101 (1970).
166. 35 U.S.C. § 102 (1970).
167. 35 U.S.C. § 103 (1970).
168. Donald v. Zack Meyer's T.V. Sales and Serv., 426 F.2d 1027 (5th Cir. 1970).