UIC John Marshall Journal of Information Technology & Privacy Law

Volume 4
Issue 2 *Computer/Law Journal - Fall 1983*

Article 1

Fall 1983

Protection of Programming in the Aftermath of Diamond v. Diehr, 4 Computer L.J. 207 (1983)

Donald E. Stout

PROTECTION OF PROGRAMMING IN THE AFTERMATH OF *DIAMOND V. DIEHR*†

by DONALD E. STOUT*

TABLE OF CONTENTS

* B.S.E.E. 1968, Pennsylvania State University; J.D. 1972, George Washington University National Law Center. Mr. Stout is associated with the law firm of Antonelli, Terry & Wands in Washington, D.C.

I. INTRODUCTION

The Supreme Court's recent decision in *Diamond v. Diehr*[1] culminated over a decade[2] of litigation on the patentability[3] of programming.[4] *Diehr*, post-*Diehr* decisions, and new U.S. Patent and Trademark Office (USPTO) guidelines on the patentability of programming have significantly clarified the types of programming which may be patented. The post-*Diehr* decisions provide a consistent body of law for assessing the patentability of inventions involving programming. The new USPTO guidelines provide a significantly broader scope of patent protection for programming than that which was previously afforded by the courts.

Programming has traditionally been protected through the mechanisms of copyright or trade secrets.[5] However, both mechanisms have had recognized deficiencies in providing effective protection for the underlying logic used for coding the program, known as the algorithm. The recent changes in the law and the USPTO's will-

1. Diamond v. Diehr, 450 U.S. 175 (1981).

2. *In re* Prater, 415 F.2d 1378 (C.C.P.A. 1968), *petition for reh'g granted*, 415 F.2d 1390 (C.C.P.A. 1969) (first significant case which addressed eligibility of inventions involving computer programming for patent protection).

3. The eligibility of inventions for patent protection arises under 35 U.S.C. § 101 (1976). Section 101 of the Patent Code defines the subject matter which is eligible for patent protection: "Whoever invents or discovers any new and useful process, machine, manufacture or composition of matter or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title." In this Article, "patentability of programming" or "patentable subject matter" will be used as a shorthand notation for identifying this issue.

4. "Programming" as used in this Article refers to the process of formulating an algorithm and a series of steps in a computer language to solve a problem. The problem to be solved may be either scientific or nonscientific in nature. The medium of expression of the resultant program may be in software or in hardware, such as a read-only memory (ROM).

5. Trade secret protection of programming typically imposes conditions of confidentiality between the buyer and seller, the lessor and lessee, or the employer and employees. Generally, a breach in confidentiality must be shown in order to obtain trade secret relief.

ingness to consider programming as patentable subject matter provide a third mechanism for protecting many types of programming. In many instances, combinations of patent, copyright and trade secret protection may now be used to protect programming more effectively than any single mechanism previously used.

II. PATENT PROTECTION

A. HISTORICAL PERSPECTIVE

Prior to the decision in *Diehr*[6], the law on the patentability of programming was constantly changing and was so unpredictable that the patent law was seldom used to protect programming even though the potential dollar loss was enormous.[7] The principal forums for the development of the law on the patentability of programming were the United States Patent and Trademark Office (USPTO), the Court of Customs and Patent Appeals (CCPA),[8] and the United States Supreme Court.

Prior to the Supreme Court's decision in *Gottschalk v. Benson*,[9] the CCPA rejected the USPTO's reliance on the "mental steps doctrine" as a basis for refusing to consider programming patentable subject matter and substituted its own expansive "technological arts" standard.[10] Under the mental steps doctrine, inventions which required human thought in whole or in part for their practice were judicially considered not to be patentable subject matter. The doctrine had its origin in attempts to patent technologies much older than digital computing. With the doctrine in mind, the USPTO argued that claim recitations[11] such as "determining," "registering,"

6. *See* Blumenthal & Riter, *Statutory or Non-Statutory?: An Analysis of the Patentability of Computer Related Inventions*, 62 J. PAT. OFF. SOC'Y, No. 8, 454–520 (1981) for a detailed review of pre-*Diehr* decisions on the patentability of programming.

7. In Parker v. Flook, 437 U.S. 584, 587 n.7 (1978), an estimate of the value of software in use in the United States in 1976 was stated to be $43.1 billion with the projection of $70.7 billion by 1980.

8. On October 1, 1982, the CCPA became the Court of Appeals for the Federal Circuit (CAFC) with appellate jurisdiction, *inter alia*, over all patent appeals from the United States Patent and Trademark Office and the federal district courts. The court will play a significant role in the continued development of all facets of the patent law on programming.

9. Gottschalk v. Benson, 409 U.S. 63 (1972).

10. *See infra* text accompanying notes 30-34.

11. 35 U.S.C. § 112 (1976) requires, *inter alia*, that the claims particularly point out and distinctly claim the subject matter which the applicant regards as his invention. The claims of a patent define the scope of the subject matter covered by the patent upon which the patentee's right of exclusion is based. All of the precedent discussed *infra* is concerned with the scope of the claimed subject matter. What appear to be small differences in the scope of the claims can be the basis for subject

"counting," "observing," and "measuring"[12] were not subject matter eligible for patent protection because the claimed acts were capable of being performed by humans.[13]

The USPTO used the mental steps doctrine as its first line of defense in refusing to grant patents on inventions involving computer programming. In large part, USPTO resistance to granting patents on programming stemmed from its admittedly inadequate resources[14] to examine patents involving programming for compliance with the statutory criteria of novelty[15] and unobviousness[16] which all patentable inventions must satisfy. Even today the USPTO does not have a large body of "reference" programming which enables it to determine adequately if an application is novel and unobvious independent of information provided by a patent applicant.

In the *Prater* opinion, the CCPA stated its rationale why programs are patentable subject matter: "No reason is now apparent to us why, based on the Constitution, statute or case law, apparatus *and* process claims broad enough to encompass the operation of a programmed general-purpose digital computer are necessarily unpatentable."[17] The court's reasoning provided the foundation for the technological arts standard developed in subsequent cases.

In *Musgrave*, the CCPA defined this new test for determining whether or not a computer was patentable subject matter in terms of whether the claimed invention related to the "technological arts."[18] The new "technological arts" test was intended to expand the scope of patentable subject matter defined by 35 U.S.C. section 101 to the constitutional limit of promoting progress in the sciences

---

matter being characterized as either patentable or unpatentable under § 101. *Compare Flook*, 437 U.S. 584 (1978) (method of updating alarm limits during catalytic conversion process held not patentable under § 101) *with Diehr*, 450 U.S. 175 (1981) (process for molding raw, synthetic rubber into cured precision products held eligible for patent protection under § 101).

12. *See In re* Abrams, 188 F.2d 165 (C.C.P.A. 1951); *see also* Halliburton Oil Well Cementing Co. v. Walker, 64 U.S.P.Q. (BNA) 278 (1944).

13. *See In re* Waldbaum, 457 F.2d 997 (C.C.P.A. 1972); *In re* Foster, 438 F.2d 1011 (C.C.P.A. 1971); *In re* Musgrave, 431 F.2d 882 (C.C.P.A. 1970).

14. The 1966 report of the President's Commission of the Patent System stated:

> The Patent Office now cannot examine applications for programs because of a lack of a classification technique and the requisite search files. Even if these were available, reliable searches would not be feasible or economic because of the tremendous volume of prior art being generated. Without this search, the patenting of programs would be tantamount to mere registration and the presumption of validity would be all but nonexistent.

15. 35 U.S.C. § 102 (1976).

16. 35 U.S.C. § 103 (1976).

17. 415 U.S. at 1403 n.29 (emphasis in original).

18. 431 F.2d at 893.

and useful arts. Judge Rich stated in his concurring opinion in *In re Waldbaum*:

> The phrase "useful arts" which was written into the Constitution conjures up images of the Franklin stove, horse collars, and buggy whips. The term "technological arts" was selected in *Musgrave* as probably having a connotation in these times roughly equivalent to that which "useful arts" had in the eighteenth century. No new legal concept was intended. In fact in *Musgrave* it was coupled with reference to the "useful arts" provision in the Constitution. Again in *In re Benson*, . . . when the same test was applied, the question asked was whether the process was not "in the technological or useful arts." Now we have come full circle in pointing out that the intention all along has been to convey the same idea and to occupy whatever ground the Constitution permits with respect to the categories of patentable subject matter named in section 101.[19]

B. PRE-*DIEHR* DECISIONS

1. Gottschalk v. Benson

The Supreme Court's decision in *Gottschalk v. Benson*[20] was a strong repudiation of the CCPA's technological arts test. The *Benson* patent application disclosed a method for converting binary coded decimal (BCD) numerals into binary numerals. The Supreme Court described the method sought to be patented as varying

> the ordinary arithmatic steps a human would use by changing the order of the steps, changing the symbolism for writing the multiplier used in some steps and by taking subtotals after each successive operation. The mathematical procedures can be carried out in existing computers long in use, no new machinery being necessary. And, *as noted they can also be performed without a computer*.[21]

The process claim before the Supreme Court read:

> A data processing method for converting binary coded decimal number representations into binary number representations comprising the steps of

---

19. 457 F.2d 997 (C.C.P.A. 1972) (citations omitted).

20. 409 U.S. 63 (1972).

21. 409 U.S. at 67 (emphasis added). In their respective decisions in *Benson*, the CCPA and the Supreme Court disagreed on the subject matter covered by the claims. Judge Rich of the CCPA found that the recitation of a reentrant shift register in claim 8 precluded any application other than a machine with a "reentrant shift register." Likewise, claim 13 was concluded to have "no practical use other than the more efficient operation and utilization of a machine known as a digital computer." Thus claims 8 and 13 were held by the CCPA to be patentable subject matter for the reason that they covered a machine or a machine performed process. *See In re* Benson, 441 F.2d 682 (C.C.P.A. 1971).

(1) testing each binary digit position '1' beginning with the least significant binary digit position, of the most significant decimal digit representation for a binary '0' or a binary '1';

(2) if a binary '0' is detected, repeating step (1) for the next least significant binary digit position of said most significant decimal digit representation;

(3) if a binary '1' is detected, adding a binary '1' at the (i + 1)th and (i + 3)th least significant binary digit positions of the next lesser significant decimal digit representation, and repeating step (1) for the next least significant binary digit position of said most significant decimal digit representation;

(4) upon exhausting the binary digit positions of said most significant decimal digit representation, repeating steps (1) through (3) for the next lesser significant decimal digit representation as modified by the previous execution of steps (1) through (3); and

(5) repeating steps (1) through (4) until the second least significant decimal digit representation has been so processed.[22]

The Court was clearly troubled by the breadth of the process claim which in its view covered both known and unknown uses of the algorithm and thus effectively covered the algorithm itself.

Here the "process" claim is so *abstract and sweeping* as to cover both known and unknown uses of the BCD to *pure binary conversion.* The end use may (1) vary from the operation of a train to verification of drivers' licenses to researching the law books for precedents and (2) be performed through any existing machinery or future-devised machinery or without any apparatus.[23]

Moreover, the Court's reference to "pure binary conversion" had a connotation similar to the mental steps doctrine which had been previously replaced by the CCPA's technological arts test as the basis for determining the patentability of programming.

The Supreme Court characterized its holding "in a nutshell":

It is conceded that one may not patent an idea. But in practical effect that would be the result if the formula for converting BCD numerals to pure binary numerals were patented in this case. The mathematical formula involved here has no substantial practical application except in connection with a digital computer, which means that *if the judgment below is affirmed, the patent would wholly preempt the mathematical formula and in practical effect would be a patent on the algorithm itself.*[24]

The "nutshell" holding of *Benson*, which proscribed preemption of

22. 409 U.S. at 74.

23. *Id.* at 68 (emphasis added). The *Benson* Court stated an algorithm to be "[a] procedure for solving a given type of mathematical problem." 409 U.S. at 65. This definition has been adopted in later decisions of the Supreme Court and the CCPA.

24. *Id.* at 71-72 (emphasis added).

mathematical formulas, is today a keystone part of the test for determining the patentability of programming.

2. In re Christensen

The CCPA extended the limited holding[25] of *Benson* in *Christensen*.[26] In *Christensen*, the invention sought to be patented was of a method for determining porosity of a subsurface formation[27] involving the solution of a novel equation on a digital computer limited to geological applications. The claimed invention recited antecedent data gathering steps in conjunction with the solution of a mathematical algorithm. The CCPA held that the recitation of the antecedent data gathering steps necessary for performing a method of calculation would not convert an unpatentable method of calculation into patentable subject matter.

> The issue before us in the instant case is also a narrow one, namely, is a method claim in which the point of novelty is a mathematical equation to be solved as the final step of the method, a statutory method? We follow the Supreme Court [in *Benson*] in concluding that the answer is in the negative. Given that the method of solving a mathematical equation may not be the subject of patent protection, it follows that the addition of the old and necessary antecedent steps of establishing values for the variable in the equation cannot convert the unpatentable method to patentable subject matter.[28]

3. In re Richman

In *Richman*,[29] the CCPA extended *Christensen* to hold that even the presence of novel and unobvious antecedent data gathering steps would not convert an otherwise unpatentable method of calculation involving a mathematical algorithm into patentable subject matter. Under *Richman*, all inventions which were properly characterizable as "methods of calculation" (algorithms as used in *Benson*) were unpatentable subject matter. In effect, the patentability of programming had evolved into an inquiry of the identity of the invention described in the patent specification, with *applications* of methods of calculation which involved more than antecedent data

25. In Dann v. Johnson, 425 U.S. 219 (1976), Justice Marshall characterized the Benson holding: "Our limited holding . . . was that respondent's method was not a patentable 'process' as that term is defined in 35 U.S.C. § 100(b)." *Id.* at 224.

26. *In re* Christensen, 478 F.2d 1392 (C.C.P.A. 1973).

27. The application disclosed that the porosity of subsurface formations is one of the parameters which are useful to geologists, petroleum engineers and others interested in analyzing lithologic formations.

28. 478 F.2d at 1394.

29. *In re* Richman, 563 F.2d 1026 (C.C.P.A. 1977).

gathering steps or limitations to particular fields of use, being patentable subject matter.

4. *Other Decisions*

Outside of the area of patent applications having claims to methods of calculation, the CCPA had little trouble construing *Benson* narrowly to find patentable subject matter. In *In re Johnson*,[30] *In re Noll*,[31] *In re Chatfield*,[32] and *In re Deutsch*,[33] the CCPA found the subject matter patentable because the claimed inventions involved subject matter not limited to methods of calculation for solving mathematical equations. In substance, the inventions in these cases were found to involve applications in the technological arts. The USPTO was distressed that the CCPA was narrowly construing the *Benson* decision.[34] The USPTO believed the CCPA was finding patentable subject matter in patent applications which were clearly unpatentable under its perception of the *Benson* decision.

5. Parker v. Flook

In *Parker v. Flook*,[35] the Supreme Court addressed the issue of patentability of a method for updating alarm limits. An alarm limit was disclosed in the patent application as a parameter in a catalytic hydrocarbon conversion process. The broadest claim was:

> 1. A method for updating the value of at least one alarm limit on at least one process variable involved in a process comprising the catalytic chemical conversion of hydrocarbons wherein said alarm limit has a current value of
>
> $$Bo + K$$
>
> wherein Bo is the current alarm base and K is a predetermined alarm offset which comprises:
>
> (1) Determining the present value of said process variable, said present value being defined as PVL;
>
> (2) Determining a new alarm base B_1 using the following equation:
>
> $$B_1 = Bo(1.0 - F) + PVL(F)$$
>
> where F is a predetermined number greater than zero and less than 1.0;
>
> (3) Determining an updated alarm limit value which is defined as

30. *In re* Johnson, 502 F.2d 765 (C.C.P.A. 1974).

31. *In re* Noll, 545 F.2d 141 (C.C.P.A. 1976).

32. *In re* Chatfield, 545 F.2d 152 (C.C.P.A. 1976).

33. *In re* Deutsch, 553 F.2d 689 (C.C.P.A. 1977).

34. *See* Diamond v. Diehr, 450 U.S. at 202 (Stevens, J., dissenting) ("The Court of Customs and Patent Appeals in subsequent cases began to narrow its interpretation of *Benson*.").

35. Parker v. Flook, 437 U.S. 584 (1978).

$B_1 + K$; and, thereafter

(4) Adjusting said alarm limit to said updated alarm limit value.[36]

The Supreme Court held that Flook's method was not patentable subject matter:

> Respondent's process is unpatentable under § 101 not because it contains a mathematical algorithm as one component but because *once that algorithm is assumed to be within the prior art, the application, considered as a whole, contains no patentable invention.* Even though a phenomenon of nature or mathematical formula may be well known, an inventive application of the principle may be patented. Conversely, the discovery of such a phenomenon cannot support a patent unless there is some other inventive concept in its application.
>
> Here it is absolutely clear that respondent's application contains no claim of patentable invention. The chemical processes involved in catalytic conversion of hydrocarbons are well known, as are the practice of monitoring the chemical process variables, the use of alarm limits to trigger alarms the notion that alarm limit values must be recomputed and readjusted, and the use of computers for "automatic process monitoring-alarming." Respondent's application simply provides a new and presumably better method for calculating alarm limit values. If we assume that the method was also known as we must under the reasoning in Morse, then respondent's claim is, in effect, comparable to a claim that the formula $2\pi r$ can be usefully applied in determining the circumference of a wheel. *As the Court of Customs and Patent Appeals has explained, "if a claim is directed essentially to a method of calculating, using a mathematical formula even if the solution is for a specific purpose, the claimed method is nonstatutory."*[37]

The factual basis for the Supreme Court's holding was its perception that the claimed invention was essentially a method of calculation for a specific purpose. The Court rejected Flook's argument that "the presence of special 'post-solution' activity—the adjustment of the alarm limit to the figure computed according to the formula—distinguishes this case from *Benson* and makes his process patentable."[38]

The Supreme Court viewed the alarm limit recited in the claims as a pure number which was divorced from an *application* to specific technology:

> The patent application does not purport to explain how to select the appropriate margin of safety, the weighting factor, or any of the other variables. Nor does it purport to contain any disclosure relat-

36. *Id.* at 596-97.

37. *Id.* at 594-95 (emphasis added).

38. *Id.* at 590.

> ing to the chemical processes at work, the monitoring of process variables, or the means of setting off an alarm or adjusting an alarm system. All that it provides is a formula for computing an updated alarm limit."[39]

Viewing Flook's disclosures as being deficient in teaching specific applications of alarm limits, the Supreme Court found *Benson* controlling in *Flook*—the only difference being the presence of post-solution activity in *Flook* which in *Richman* was held *not* to make an invention involving programming patentable subject matter. In a footnote the Supreme Court characterized its holding: "Very simply, our holding today is that a claim for an improved method of calculation, even when tied to a specific end use, is unpatentable subject matter under § 101."[40] The Supreme Court's test in *Flook* "that [the] algorithm is assumed to be within the prior art" provided a basis for the USPTO to find most forms of programming unpatentable subject matter and was soundly criticized as mixing considerations of novelty and obviousness with considerations of statutory subject matter.[41]

6. In re Freeman *and* In re Walter

In response to the *Benson* and *Flook* cases, the CCPA developed a two-part test for determining patentability of programming. The two-part test was based upon the holding in *Benson* that claims which "preempt" methods of calculation involving mathematical algorithms are not patentable subject matter. In *In re Freeman*,[42] the invention sought to be patented was a computer controlled phototypesetting machine. The hardware was of conventional design but the computer control program produced a machine which was undisputedly new and obvious. The USPTO argued in *Freeman* that the fact that the only disclosed novelty in the specification was the program precluded a finding of patentable subject matter. The CCPA's test required (1) determination whether the *claimed* invention directly or indirectly recites an algorithm and (2) that if an algorithm is found, it must be determined whether the claim wholly preempts the algorithm, that is, whether preemption precludes patentable subject matter. Application of the two-part test in *Freeman* resulted in reversal of the USPTO rejection.

---

39. *Id.* at 586.

40. *Id.* at 595 n.18.

41. *See* Blumenthal & Riter, *supra* note 6, at 487: "It is unfortunate that the Supreme Court analysis dictates placing the mathematical algorithm in the 'prior art.' The term 'prior art' imports a novelty/obviousness question into the picture and tends to confuse issues."

42. *In re* Freeman, 573 F.2d 1237 (C.C.P.A. 1978).

In response to the *Flook* Court's holding that methods of calculation applied to find particular end uses (trivial post-solution) were not patentable subject matter, the CCPA modified its *Freeman* test in *In re Walter*[43] to not require preemption of the algorithm to find unpatentable subject matter:

> Once a mathematical algorithm has been found, the claim *as a whole* must be further analyzed. If it appears that the mathematical algorithm is implemented in a specific manner to define structural relationships between the physical elements of the claim (in apparatus claims) or to refine or limit claim steps (in process claims), the claim being otherwise statutory, the claim passes muster under § 101. If, however, the mathematical algorithm is merely presented and solved by the claimed invention, as was the case in *Benson* and *Flook*, and is not applied in any manner to physical elements or process steps, no amount of post-solution activity will render the claim statutory: nor is it saved by a preamble merely reciting the field of use of the mathematical algorithm.[44]

In *Walter*, the CCPA sustained the USPTO rejection that the claimed signals were not limited to physical quantities:

> Claim 12, and claims 10 and 11 dependent therefrom, are not presented in Jepson format, but they suffer from a fundamental flaw which places them outside the bounds of § 101. *These claims are directed to the process of cross-correlation in the abstract.* They are not limited to any particular art or technology, unless pure mathematics is considered as an art or technology. The "signals" processed by the inventions of claims 10-12 may represent either physical quantities or abstract quantities; the claims do not require one or the other. The claims thus dominate the particular method of cross-correlation in any and all arts. They are *classic examples of an attempt to embrace the algorithm or scientific truth itself rather than a particular application*.[45]

The *Freeman-Walter* test defined the metes and bounds of the patentability of programming in terms of whether the claims implemented the mathematical algorithm either in structural relationships between the physical elements or the refining or limiting of process steps. Any claim which recited a mathematical algorithm as an "implementation" in structural relationships or process steps was patentable subject matter. Any claim which did not implement the algorithm, such as a method of calculation, was not patentable subject matter.

The USPTO viewed the *Freeman-Walter* test as erroneous in view of the *Flook* decision. The statements in *Flook* that (1) the al-

43. *In re* Walter, 618 F.2d 758 (C.C.P.A. 1980).

44. *Id.* at 767.

45. *Id.* at 770 (emphasis added).

gorithm is to be assumed in the prior art, and that (2) there is no patentable subject matter "unless there is some other inventive concept in its application,"[46] was read by the USPTO as a broad test which precluded patentability for most types of programming—a position consistent with its long standing resistance to the patenting of programming.

The CCPA's *Freeman-Walter* test was broadly expansive of the forms of programming which were patentable subject matter. All forms of programming were patentable subject matter except methods of calculation (1) involving mathematical algorithms including those limited to particular fields of use; (2) involving mathematical algorithms in conjunction with trivial post solution activity; or (3) involving mathematical algorithms in conjunction with the gathering of necessary data to perform the solution. On the other hand, the USPTO test, which was based upon its reading of *Flook*, was totally centered on the nonalgorithmical parts of the claimed invention, which were required to be novel and unobvious implementations of the mathematical algorithm to be patentable subject matter. The stage for collision of these diverse viewpoints on the patentability of programming was thus set for judicial resolution in *Diehr*.

C. The *Diehr* Decision

The invention at issue in *Diehr* was a computer-controlled process for curing rubber. The curing of rubber in accordance with the *Diehr* invention involved the precision heating of rubber in a heating mold for an optimum curing time which was *calculated* by a programmed digital computer in accordance with the *known* Arrhenius equation.[47] Variations in mold temperature were continually monitored by a thermocouple to provide real time information of mold performance to update calculation of the optimum mold cure time. The mold was opened when the actual *elapsed* curing time

---

46. 437 U.S. at 594-95.

47. In a footnote the Supreme Court described the Arrhenius equation as follows:

> The equation is named after its discoverer Svante Arrhenius and has long been used to calculate the cure time in rubber molding presses. The equation can be expresses [sic] as follows:
>
> $$V = CZ + X$$
>
> Wherein ln v is the natural logarithm of v, the total required cure time; C is the activation constant, a unique figure for each batch of each compound being molded, determined in accordance with rheometer measurements of each batch; Z is the temperature in the mold; and x is a constant dependent on the geometry of the particular mold in the press. A rheometer is an instrument to measure flow of viscous substances.

450 U.S. at 177 n.2.

matched the calculated optimum cure time.[48]

The USPTO viewed *Diehr* as a perfect vehicle to reaffirm the *Flook* holding and to obtain reversal of the CCPA's narrow *Freeman* test. The broadest claim of the invention sought to be patented in *Diehr* was:

> 1. A method of operating a rubber-molding press for precision molded compounds with the aid of a digital computer, comprising:
>
> providing said computer with a data base for said press including at least,
>
> natural logarithm conversion data (ln),
>
> the activation energy constant (C) unique to each batch of said component being molded, and
>
> a constant (x) dependent upon the geometry of the particular mold of the press,
>
> initiating an interval timer in said computer upon the closure of the press for monitoring the elapsed time of said closure,
>
> constantly determining the temperature (Z) of the mold at a location closely adjacent to the mold cavity in the press during molding,
>
> constantly providing the computer with the temperature (Z),
>
> repetitively calculating in the computer, at frequent intervals during each cure, the Arrhenius equation for reaction time during the cure, which is
>
> $$\ln v = CZ + x$$
>
> where v is the total required cure time, repetitively comparing in the computer at said frequent intervals during the cure each said calculation of the total required cure time calculated with the Arrhenius equation and said elapsed time, and
>
> opening the press automatically when a said comparison indicated equivalence.[49]

Application of the *Walter* test dictated that Diehr's invention *was* patentable subject matter because there was no preemption of the Arrhenius equation. As a whole, the claimed invention was an application of a method of calculation involving a mathematical algorithm. The manifestation of the application of the algorithm was found in a number of the claimed steps, such as "providing," "constantly determining," "constantly providing," and "opening the press," which were not part of the calculation of the Arrhenius equation. Application of the *Flook* test by the court, however, dictated

48. The government did not dispute the applicant's statement that previous rubber curing processes were inadequate. *See id.* at 178 n.4. In patent infringement litigation, the government's failure to challenge the applicant's assertion that the invention solved an industry-wide problem would be a serious disadvantage in proving that a patent was invalid.

49. *Id.* at 179 n.5.

that the claimed process was not patentable subject matter. In *Diehr*, no serious challenge was made to the USPTO Examiner's assertion that "the remaining steps—installing rubber in the press and the subsequent closing of the press—were 'conventional' and necessary to the process and cannot be the basis for patentability."[50] Thus, once the focus shifted to the nonalgorithmic steps, which were admitted to be old, a finding of unpatentable subject matter under *Flook* was bound to follow.

In the USPTO's view, the *only* difference between *Diehr* and *Flook* was the antiquity of the Arrhenius equation in *Diehr* compared to the novelty of *Flook's* algorithm. Thus, given the facts in *Diehr*, the USPTO reasoned that the Supreme Court would necessarily reaffirm *Flook*. The USPTO did not forsee any arguments which would warrant a finding of patentable subject matter under the *Flook* test.

The *Diehr* holding *sub silentio*[51] reversed *Flook* and fundamentally changed the patentability of programming. Stated broadly, the *Diehr* holding was that *applications* of nonpatentable subject matter, such as laws of nature and mathematical algorithms, were patentable subject matter.[52] The Supreme Court's holding in *Diehr* was

---

50. *Id.* at 180-81.

51. *Id.* at 189 n.12. The majority *Diehr* opinion attempted to avoid the argument that *Diehr* was a reversal of *Flook*:

> It is argued that the procedure of dissecting a claim into old and new elements is mandated by our decision in *Flook* which noted that a mathematical algorithm must be assumed to be within the "prior art." It is from this language that the [Government] premises its argument that if everything other than the algorithm is determined to be old in the art, then the claim cannot recite statutory subject matter. The fallacy in this argument is that we did not hold in *Flook* that the mathematical algorithm could not be considered at all when making the § 101 determination. To accept the analysis proffered by the [Government] would, if carried to its extreme, make all inventions unpatentable because all inventions can be reduced to underlying principles of nature which, once known, make their implementation obvious. The analysis suggested by the Government would also undermine our earlier decisions regarding the criteria to consider in determining the eligibility of a process for patent protection.

Id. at 189.

52. The *Diehr* Court stated the following:

> Our earlier opinions lend support to our present conclusion that a claim drawn to subject matter otherwise statutory does not become nonstatutory simply because it uses a mathematical formula, computer program or digital computer. In *Gottschalk v. Benson* we noted, "It is said that the decision precludes a patent for any program, servicing a computer. We do not so hold." 409 US at 71. Similarly, in *Parker v. Flook* we stated that "a process is not unpatentable simply because it contains a law of nature or a mathematical algorithm." 437 U.S. at 590. It is now commonplace that an *application* of a law of nature or mathematical formula to a known structure or process may well be deserving of patent protection. (Citations omitted, emphasis in original.)

consistent with the *Freeman-Walter* test. Although the *Freeman-Walter* test is arguably narrower than *Diehr's* application test for determining where subject matter is patentable, it is not inconsistent with *Diehr*. Under *Diehr*, an application of a method of calculation "to subject matter otherwise statutory" qualifies an invention as patentable subject matter. In the context of section 101, a "process" or "machine" which utilizes a method of calculation in an application is patentable subject matter *per se*. The determination of the boundary line between a patentable application of a mathematical algorithm and an unpatentable method of calculation involving the mathematical algorithm is determined from the consideration of the claim "as a whole."[53] While claim analysis is complicated, experienced patent practioners routinely analyze claims for compliance with sections 102, 103, and 112 of the Patent Code and thus, should not have great difficulty in applying the *Diehr* test. In the final analysis, the real difference between the *Flook* and *Diehr* inventions was in the claimed subject matter.[54] The Supreme Court did not consider Flook's updated alarm limit to be an application (a debatable point) of the calculation of the Arrhenius equation. Diehr's process

---

As Mr. Justice Stone explained four decades ago:

> While a scientific truth, or the mathematical expression of it, is not a patentable invention, a novel and useful structure created with the aid of knowledge of scientific truth may be. *Mackay Radio Corp. & Telegraph Co. v. Radio Corp. of America*, 306 U.S. 86, 94 (1939).
>
> We think this statement in *Mackay* takes us a long way toward the correct answer in this case. Arrhenius' equation is not patentable in isolation, but when a process for curing rubber is devised which incorporates in it a more efficient solution of the equation, that process is at the very least not barred at the threshold by § 101.
>
> In determining the eligibility of respondent's claimed process for patent protection under § 101, *their claims must be considered as a whole. It is inappropriate to dissect the claims into old and new elements and then to ignore the presence of the old elements in the analysis.* This is particularly true in a process claim because a new combination of steps in a process may be patentable even though all the constituents of the combination were well known and in common use before the combination was made. *The "novelty" of any element or steps in a process, or even of the process itself, is of no relevance in determining whether the subject matter of a claim falls within the § 101 categories of possibly patentable subject matter.*

450 U.S. at 187-89 (emphasis added).

53. This statement effectively overrules the *Flook* test which required the bifurcation of the claims into algorithmic and nonalgorithmic parts.

54. 450 U.S. 175, 210 n.32 (1981) (Stevens, J., dissenting) ("Indeed, the most significant distinction between the invention at issue in *Flook* and that at issue in this case lies not in the characteristics of the inventions themselves, but rather in the drafting of the claims.").

was viewed as a rubber curing process which had always been a statutory process regardless of the process steps used to practice it.

The *Diehr* decision has significantly clarified the law in several areas:

1. The proper test for determining the patentability of programming requires analysis of the claims "as a whole." The USPTO interpretation of *Flook*, which required dissection of the claims into algorithmic and nonalgorithmic parts, was overruled.

2. Applications of methods of calculation involving mathematical algorithms are patentable subject matter.

3. Methods of calculation of mathematical algorithms alone or limited to fields of use are not patentable subject matter.

4. Token post-solution activity does not transform an unpatentable method of calculation involving a mathematical algorithm into patentable subject matter.[55]

5. *Diehr* did not decide the issue of the patentability of programming involving nonmathematical algorithms. *Diehr* did not state any dicta inconsistent with nonmathematical applications of programming being patentable subject matter. The patentability of programming which does not involve mathematical algorithms has been significantly effected by new USPTO guidelines which will be discussed below.

III. THE USPTO RESPONSE TO *DIEHR*

The USPTO promulgated guidelines[56] to implement the *Diehr* decision which for the first time in its history voiced an administrative policy in favor of patents on most types of programs. Most of the guidelines define the lines of demarcation between unpatentable methods of calculation and patentable applications of methods of calculation by direct quotation from the salient parts of *Diehr*. The guidelines further conclude that "[t]he Court's requirement that the 'claims must be considered as a whole' in effect leaves viable the CCPA's two step procedure set forth in *In re Freeman* and *In re Walter*."[57]

What is truly significant about the USPTO guidelines is that they instruct the USPTO examining corps that computer programs which do *not* involve mathematical algorithms, are patentable subject matter if they otherwise do not involve other nonstatutory sub-

55. 450 U.S. at 192 n.14.

56. MANUAL OF PATENT EXAMINING PROCEDURE, SECTION 2110: PATENTABLE SUBJECT MATTER—MATHEMATICAL ALGORITHMS OR COMPUTER PROGRAMS 538-38.3 (October 1981) [hereinafter cited as USPTO Guidelines].

57. *Id.*

ject matter.[58]

> In accordance with the two-step procedure outlined above, *claims seeking coverage for a computer program would be non-statutory* under 35 U.S.C. § 101, *only if, when considered as a whole, they merely recite a mathematical algorithm, or a method of calculation.* Such an approach is the same as that contemplated for apparatus claims by the CCPA in *In re Bradley* and *Franklin*. . . .[59]

Thus, programming which does *not* involve mathematical algorithms or methods of calculation is now considered to be patentable subject matter.

Given the longstanding opposition of the USPTO to patents on software, it is truly remarkable that the agency voluntarily changed its position on the patentability of software in areas not dictated by *Diehr*. There was nothing in the *Diehr* holding which mandated USPTO change of its policy against patents on programming outside the patentability of applications of mathematical algorithms. In effect, the USPTO unilaterally and significantly changed the law where Congress refused to legislate throughout the 1970's.[60]

IV. CCPA RESPONSES TO *DIEHR*

The CCPA interpretation of the *Diehr* decision is revealed by its recent decisions in *In re Taner*,[61] *In re Pardo*,[62] *In re Abele*,[63] and *In re Meyer*.[64]

A. *IN RE ABELE*

In *Abele*, the CCPA broadened its *Freeman-Walter* test to be consistent with the "applications" standard set forth in *Diehr*. The invention was a computerized axial tomography which is known by the acronym CAT scan. Abele discovered that a useful CAT scan could be obtained while subjecting the patient to a lower level of x-ray exposure. A weighting function was used in otherwise conventional CAT scan calculations to eliminate "artifacts."

---

58. The U.S. Patent and Trademark Office has always held that certain types of subject matter other than programming are not patentable subject matter. For example, printed matter and methods of doing business are considered unpatentable subject matter under 35 U.S.C. § 101.

59. USPTO Guidelines, *supra* note 56, at 538.2.

60. In *Flook*, the Supreme Court stated that Congress has yet to decide the "[d]ifficult questions of policy concerning the kinds of programs that may be appropriate for patent protection and the form and duration of such protection." 437 U.S. at 595.

61. *In re* Taner, 681 F.2d 787 (C.C.P.A. 1982).

62. *In re* Pardo, No. 81-619 (C.C.P.A. Aug. 5, 1982).

63. *In re* Abele, No. 81-618 (C.C.P.A. Aug. 5, 1982).

64. *In re* Meyer, No. 82-510 (C.C.P.A. Sept. 16, 1982).

The USPTO Board of Appeals relied upon the *Freeman-Walter* test to sustain the Examiner's rejection of the claimed invention as not being patentable subject matter:

> When the claims are analyzed in (the manner dictated by *Walter*), it is manifest that the mathematical algorithm is not implemented in a manner to define structural relationships between physical elements in the apparatus claims or to refine or limit claim steps in the process claims. The claims do no more than present and solve a mathematical algorithm and are manifestly nonstatutory.[65]

In its decision in *Abele*, the CCPA broadened its *Freeman-Walter* test to the "applications" test of *Diehr* as the standard for determining the patentability of subject matter involving mathematical algorithms:

> [T]he *Walter* analysis . . . does not limit patentable subject matter only to claims in which structural relationships or process steps are defined, limited or refined by the application of the algorithm.
>
> Rather, *Walter* should be read as requiring no more than that the algorithm be "applied *in any manner* to physical elements or process steps," *provided that its application is circumscribed by more than a field of use limitation or non-essential post-solution activity.* Thus, if the claim would be "otherwise statutory," *id.*, albeit inoperative or less useful without the algorithm, the claim likewise presents statutory subject matter when the algorithm is included. This broad reading of *Walter*, we conclude, is in accord with the Supreme Court decisions.[66]

Under *Abele*, then, programs involving mathematical algorithms are patentable subject matter *unless* the claimed invention preempts the algorithm, is a method of calculation limited only to a field of use or is a method of calculation involving nonessential post-solution activity.[67]

The application of these principles is illustrated by the CCPA's treatment of Abele's claims, only some of which the court found eligible for patent protection. Claims five and six are representative:

> 5. A method of displaying data in a field comprising the steps of calculating the difference between the local value of the data at a data point in the field and the average value of the data in a region of the field which surrounds said point for each point in said field, and displaying the value of said difference as a signed gray scale at a point in a picture which corresponds to said data point.
>
> 6. The method of claim 5 wherein said date is x-ray attenuation

65. *Quoted in Abele*, 684 F.2d at 905.

66. *Id.* at 906 (emphasis added).

67. Presumably, "non-essential post-solution activity" involves subject matter such as the updating of the alarm limit in *Flook*. Precise delineation of the metes and bounds of "non-essential post-solution activity" will await further decision.

> data produced in a two dimensional field by a computed tomography scanner.[68]

Claim five was found to present "no more than the calculation of a number and display of the result, albeit in a particular format."[69] It was "directed solely to the mathematical algorithm portion of appellants invention and is, thus, not statutory subject matter under § 101."[70]

In contrast, the method of claim six required "X-ray attenuation data," a difference the court found significant:

> The specification indicated that such attenuation data is available only when an x-ray beam is produced by a CAT scanner, passed through an object, and detected upon its exit. Only after these steps have been completed is the algorithm performed, and the resultant modified data displayed in the required format.[71]

The CCPA recognized that the *Freeman-Walter* test requirement of refinement or limitation of the earlier claim steps was not present in claim six. Use of the applications test of *Diehr*, however, which focuses on the claimed process as a whole for determination if otherwise statutory subject matter is present, resulted in the following finding:

> [P]roduction, detection, and display steps . . . [are] manifestly statutory subject matter . . . . What appellants have done is to discover an *application* of an algorithm to process steps which are themselves part of an overall process which is statutory. Hence, claim six cannot be construed as a mere procedure for solving a given mathematical problem.[72]

B. *In re Pardo*

Pardo[73] involved a compiler program for executing program steps out of the order in which they were presented. The *Pardo* decision is important because it involved programming subject matter which does *not* involve mathematical algorithms. Claim thirty is representative of the subject matter sought to be patented:

> 30. A process of operating a general purpose data processor of known type to enable the data processor to execute formulas in an object program comprising a plurality of formulas, such that the same results will be produced when using the same given data, re-

68. 684 F.2d at 908.
69. *Id.* at 909.
70. *Id.* at 908-09.
71. *Id.*
72. *Id.* at 909 (emphasis in original).
73. *In re* Pardo, 684 F.2d 912 (C.C.P.A. 1982).

> gardless of the sequence in which said formulas are presented in said object program, comprising the steps of:
>
> (a) examining each of said formulas in a storage area of the data processor to determine which formulas can be designated as defined;
>
> (b) executing, in the sequence in which each formula is designated as defined, said formulas designated as defined;
>
> (c) repeating steps (a) and (b) for at least undefined formulas as many times as required until all said formulas have been designated as defined and have been executed;
>
> whereby to produce the same results upon execution of the formulas in the sequence recited in step (b) when using the same given data, regardless of the order in which said formulas were presented in the object program prior to said process.[74]

The USPTO Examiner acknowledged that the algorithm was not mathematical.

The CCPA rejected the USPTO solicitor's argument that the "claims *indirectly* amount to mathematical calculations, because the programs [disclosed in the patent application] subjected to appellants' process are exemplified as mathematical formulae."[75] The CCPA reiterated the broad scope of section 101, which is to be limited only by judicial exceptions:

> "[A] claim drawn to subject matter otherwise statutory does not become nonstatutory simply because it uses a mathematical formula, computer program, or digital computer." *Diamond* v. *Diehr*, *supra* at 187, ... Indeed, any process, machine, manufacture, or composition of matter constitutes statutory subject matter *unless* it falls within a judicially determined exception to section 101 . . . . The appealed claims do not fall within any such exception.[76]

C. *In re Meyer*

In *Meyer*,[77] the invention was a computer program for analyzing the response of a complex system, such as the human nervous system, to a series of tests. The program predicted the probability of function or malfunction of the elements in the system being tested. The invention in essence implemented the analysis function of a neurologist in a computer program.[78] The CCPA held that the

---

74. *Id.* at 913.

75. *Id.* at 916 (emphasis in original).

76. *Id.* (emphasis in original).

77. *In re* Meyer, 688 F.2d 789 (C.C.P.A. 1982).

78. The U.S. Patent and Trademark Office Solicitor characterized the invention as a "diagnostic" or "memory" aid for a physician and emphasized that the invention does not conduct a diagnosis in and of itself, but is used by a doctor when performing

claimed invention[79] was in essence a mathematical algorithm of a mental process, which is not patentable subject matter. The meaning of the term "mathematical algorithm" was discussed in detail:

> Scientific principles, such as the relationship between mass and energy, and laws of nature, such as the acceleration of gravity, namely, a = 32 ft./sec.^{2}, can be represented in mathematical format. However, some mathematical algorithms and formulae do not represent scientific principles or laws of nature; they represent ideas or mental processes and are simply logical vehicles for communicating possible solutions to complex problems. The presence of a mathematical algorithm or formula in a claim is merely an indication that a scientific principle, law of nature, idea or mental process *may* be the subject matter claimed and, thus, justify a rejection of that claim under 35 U.S.C. § 101; but the presence of a mathematical algorithm or formula is only a signpost for further analysis.[80]

The CCPA's characterization of mathematical algorithms as "simply logical vehicles for communicating possible solutions to complex problems" delineates those subject areas of programming which may be judicially *excepted* from being patentable subject matter under section 101.

---

a diagnosis to store and to accumulate test responses obtained by this standard process of elimination and to narrow the neurological area of possible malfunction. In fact, the Solicitor indicated that these standard tests have been employed for many years and that the more experienced the doctor and the better his memory, the less would be his need (if any) for this invention. *Id.*

79. Claim one is representative of the invention sought to be patented.

> 1. A process for identifying [sic] locations of probable malfunction in a complex system, said process comprising the steps of:
>
> (a) selecting a plurality of elements in the complex system, said elements having known locations;
>
> (b) initializing a factor associated with each of said elements;
>
> (c) testing the complex system for a response, which response, if effective, requires proper functioning of certain said elements, the probable identity [sic] of at least some of these certain elements being known;
>
> (d) determining whether said response of the complex system was at least partially effective or ineffective;
>
> (e) modifying the factor associated with at least some of said elements known to be possible [sic] involved in the response in accordance with the effectiveness of the response; and
>
> (f) repeating steps (c), (d) and (e) for further responses of the complex system to obtain resultant factors for at least some of said elements,
>
> whereby said resultant factors are indicative of probable malfunction of their associated elements and thereby indicative of probable malfunction at the locations of these elements.

Id. at 792-93.

80. *Id.* at 794-95 (emphasis in original).

D. *IN RE TANER*

The applicant in *Taner*[81] sought patent protection for a method of seismic exploration in which spherical seismic waves are used during exploration to simulate the earth's reponse to plane or cylindrical waves. The claimed invention[82] utilizes a mathematical algorithm implemented on a digital computer which sums reflection signals of the spherical waves.

The USPTO Board of Appeals held that *Christensen*[83] was controlling for the reason that the claimed invention was a method of calculation with the antecedent stops merely serving to supply the required data for solving the mathematical algorithm. Under *Christensen* and *Richman*,[84] the presence of data gathering steps in the claims was not considered sufficient to convert claims involving a method of calculation utilizing a mathematical algorithm into a patentable process.

The CCPA viewed the claimed signal as a physical conversion of the spherical seismic signals. It reasoned that the expression of physical apparitions (the claimed signals) in mathematical terms was irrelevant in controlling the patentability protection.[85] The CCPA expressly overruled *Christensen* to the extent that it conflicted with the applications test of *Diehr*:

> [W]hen a claim containing a mathematical formula implements or applies that formula in a structure or process, which, when considered as a whole, is performing a function which the patent laws were designed to protect (e.g., transforming or reducing an article

---

81. *In re* Taner, 681 F.2d 787 (C.C.P.A. 1982).

82. Claim one is representative of the claimed invention:

> 1. A method of seismic exploration by simulating from substantially spherical seismic waves the reflection response of the earth to seismic energy having a substantially continuous wavefront over an extent of an area being explored having at least one dimension which is large relative to a seismic wavelength, comprising the steps of:
>
> (a) imparting the spherical seismic energy waves into the earth from a seismic source at a source position;
>
> (b) generating a plurality of reflection signals in response to the seismic energy waves at a set of receiver positions spaced in an array over an extent having at least one dimension which is large relative to a seismic wave length; and
>
> (c) summing the reflection signals to form for the source position a signal simulating the reflection response of the earth to seismic energy having a substantially continuous wavefront over at least one dimension which is large relative to a seismic energy wavelength.

Id. at 788.

83. *In re* Christensen, 478 F.2d 1392 (C.C.P.A.1973).

84. *In re* Richman, 563 F.2d 1026 (C.C.P.A. 1977).

85. The characterization of the signals as physical apparitions was supported by *In re* Sherwood, 613 F.2d 809 (C.C.P.A. 1980) and *In re* Johnson, 589 F.2d 1070 (C.C.P.A. 1978).

> to a different state or thing), then the claim satisfies the requirements of § 101. . . . Accordingly, to the extent that it conflicts with what we say here, *Christensen* is overruled.[86]

In other words, the recitation of antecedent data gathering steps *can* be a sufficient basis to constitute a patentable *application* of a non-patentable mathematical algorithm.

V. TYPES OF PATENTABLE PROGRAMMING

In the last two and one-half years, the Supreme Court in *Diehr*, the CCPA in *Abele*, *Pardo*, *Meyer* and *Taner*, and the USPTO have each decisively acted to expand the types of programming which are patentable subject matter. The effect of these actions is that for the first time many areas of programming are clearly patentable subject matter. The previous uncertainty about patentability has been removed for most types of programming.

There are two main categories of computer programs: system programs and application programs. A system program is used to manage or control the operation of a computing system to facilitate its use by the user. Most system programs are "transparent" to the user in that the user normally does not write the program for a specific use. An application program performs a specified task in accordance with a user's program, and its content is under the control of the user. Common examples of system programs include:

> (1) Operating Systems—or supervisory executive programs which control and allocate the computer resources of main memory and input and output peripheral devices and the execution of programs;
> (2) Assembler Programs—programs which prepare machine language programs from symbolic (human intelligible) programming language programs. Assembler programs usually translate one source code program statement to one machine language instruction. These are known as low level languages.
> (3) Compiler Programs—programs that prepare machine language programs from source programs written in a higher level programming language. A compiler program usually translates one source code program statement into more than one machine language instruction. These are known as higher level languages. The compiler program may include or perform the functions of an assembler program also.[87]

Applications programs span a wide field, but for purposes of determination of patentable subject matter, they may be catagorized as (1) programs which implement mathematical algorithms, or

---

86. 681 F.2d at 791 (citation omitted).

87. *See* Prasinos, *Legal Protection of Software Via Copyright*, 8 APLA Q.J. 252, 255 (1980).

(2) programs which do *not* implement mathematical algorithms. The patentability of applications programs involving mathematical algorithms is determined by the precedent and USPTO guidelines discussed earlier. While the issue of patentability was not before the court, *Synercom Technology, Inc. v. University Computing Co.*,[88] involved a dispute over a structural analysis program which was essentially a method of calculation involving various mathematical algorithms. The *Synercom* program was found not to be patentable subject matter.[89]

The USPTO guidelines, which permit patents on programming involving nonmathematical algorithms, will increase the number of patents issued on nonscientific applications of programming. Once it becomes generally understood that patents on nonmathematical applications of software can be obtained, patent protection will be widely considered as an additional protection mechanism in any plan for maximum security of software. The way in which patents, copyrights and trade secrets affect the protection of programming is discussed later.

The patentability of systems programs was firmly established by the CCPA in *In re Bradley*.[90] The invention in *Bradley* related to the altering or repositioning of information in a system base of a computer. The invention implemented the manipulation of information by means of a program stored on a "firmware" module such as a ROM. The USPTO Board of Appeals reasoned that the theoretical mathematical operations of a computer made systems programs unpatentable:

> In whatever form the instructions employed in appellants' invention are characterized, numerical or otherwise, we think it is accurate to say that the operation of appellants' structure is mathematical and that the instructions constitute a procedure which is algorithmic in character . . . within the definition of algorithm supplied in *Benson* and reiterated in *Flook*.[91]

Applying the *Freeman* test discussed earlier, the CCPA rejected the Board of Appeals reasoning:

> When we examine appellants' invention as a whole under the first

88. Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1003 (1978).

89. Examples of applications programs which do not involve methods of calculation utilizing mathematical algorithms are Merrill Lynch, Pierce, Fenner and Smith's *Securities Brokerage-Case Management Systems*, which is disclosed in U.S. Patent No. 4,346,442, and Frederic C. Tower's *Securities Valuation System*, which is disclosed in U.S. Patent No. 4,344,270.

90. *In re* Bradley, 600 F.2d 807 (1979), *aff'd per curiam by an equally divided court*, 209 U.S.P.Q. 97 (1981).

91. 600 F.2d at 811.

> step of this test, including the information microprogrammed into the firmware element as depicted in Figs. 14(a-i) and 15(b-c), we fail to detect the presence of any mathematical algorithm. In altering information in the system base as desired, certain "calculations" are made, such as determining whether a given quantity is equal to zero, or, as noted by the solicitor, multiplying an address in memory by sixteen to arrive at another address. However, it certainly cannot be said that comparing with zero or multiplying by sixteen is preempted by appellant's claims.[92]

The same result is reached by applying the newer *Abele* test since no mathematical algorithm is present.

The very nature of a system program—regardless of its specific function—does not involve mathematical algorithms as that term has been defined in the relevant decisions. In some way a system program modifies or controls the operation of computer hardware which is manifestly patentable subject matter under section 101 as construed by *Diehr*. Properly drafted patent claims to any system program will recite hardware that is manipulated by the program without the application of a mathematical algorithm and, *a fortiori*, will involve "subject matter otherwise statutory" as described in *Diehr*.[93] The claims in patent applications on systems programs should not even reach the threshold question of the *Freeman-Walter-Abele* tests of whether a mathematical algorithm is present. Thus, systems programs are patentable subject matter *per se*.

VI. PROSPECT FOR CHANGE IN THE LAW OF PROGRAMMING PATENTABILITY

A number of factors militate against substantial change in the law of programming patentability in the immediate future. First, the new CAFC has exclusive subject matter jurisdiction over appeals arising under the patent laws because of refusal of the USPTO to grant a patent on a pending application (formerly the CCPA's jurisdiction). In addition, the CAFC now has subject matter jurisdiction of all patent appeals arising from infringement actions in the federal district courts. The precedents of the CCPA for judging the patentability of programming will be fully controlling in all future actions of the CAFC. Second, the prospect of change in CAFC rulings on the patentability of programming is low because prior CCPA precedent which is now well-developed will be applied to all patent appeals. There is little likelihood of divergent opinions on the patentability of programming since all appeals on the subject will be processed

92. *Id.* at 813.

93. *See supra* text accompanying notes 50-53.

through the CAFC. There are no appeals involving the patentability of programming presently pending before the CAFC. Third, new USPTO guidelines which bar the rejection of most forms of programming greatly diminish the likelihood of appeals from the USPTO to the CAFC. USPTO pressure on patent examiners to examine large numbers of patent applications is a strong deterrent to the rejection of applications for failure to claim patentable subject matter.[94] Fewer rejections of applications for containing unpatentable programming necessarily will result in fewer appeals to the CAFC which in turn will create less pressure toward change in the law. It is also known that the Department of Justice has lost interest in seeking further *certiorari* petitions in an attempt to "clarify" the law as was the case with *Diehr*. Finally, Congressional action changing the scope of programming which is patentable subject matter is unlikely given the extreme political pressure which would be incident to any legislative proposals to change the status quo. In the past, Congress has refused to take action even though the Supreme Court has suggested that legislation was warranted to resolve the patentability of programming.[95]

VII. COPYRIGHT PROTECTION OF PROGRAMMING

A. Subject Matter

A review of the component steps of the programming process is helpful in examining the scope of protection afforded to programming by copyrights. The typical program begins as an "idea" in the mind of the programmer, is next reduced to a flow chart which sets forth the basic algorithm of the program, and which finally is reduced to program by coding the algorithm in a computer language. The coding process is the point in the formulation of a program at which copyright protection may be obtained.

Section 102(b) of the Copyright Act[96] clearly establishes the principle, well-established by prior case law,[97] that a copyright cov-

94. To maintain a satisfactory performance rating, the U.S. Patent and Trademark Office Examiners are required to examine a "quota" of patent applications. Failure to achieve at least 90% of the quota is a basis for an unsatisfactory rating which depending on circumstances can lead to disciplinary action such as down grading on the civil service scale or dismissal. The net effect of the quota system is that patent examiners are discouraged from rejecting applications for failure to claim patentable subject matter under § 101 because these cases traditionally require extra time to complete examination due to the greater likelihood of appeals.

95. Gottschalk v. Benson, 409 U.S. 63, 73 (1972) ("considerable problems are raised which only committees of Congress can manage").

96. 17 U.S.C. § 102(b) (1982).

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

ers the author's expression of an idea, but not the underlying idea itself: "In no case does copyright protection for an original work of authorship extend to any ideas, 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."[98] The algorithm represents the "concept" of the program, and no copyright protection may adhere to it. The actual coding, regardless of its form of recording, i.e., a piece of paper or magnetic tape, satisfies the further statutory requirement that copyrightable subject matter must be "fixed in any tangible medium of expression."[99]

B. Rights Protected By Copyright

The rights conferred by copyright which are applicable to programs are:

1. To reproduce the copyrighted work in copies or phonorecords.
2. To prepare derivative works based upon the copyrighted work.
3. To distribute copies or phonorecords of the copyrighted work to the public by sale or other transfer of ownership or by lease or lending. . . .[100]

A copyright on a program provides protection against unauthorized copying of all or a significant part of the program (derivative work) as well as unauthorized distribution of copies. A copyright does not proscribe the independent derivation of a similar program even when the underlying algorithm embodied in the copyrighted program, as distinguished from the program itself, is used as a reference for developing a program. A copyright cannot prevent "reverse engineering" of a program to derive its underlying algorithm which is used as a reference for independent derivation of new coding embodying the algorithm. Thus, exclusive reliance on copyright to protect programming will not provide sufficient statutory protection where the underlying algorithm is sufficiently novel and unobvious to warrant patent protection.[101]

---

98. *See supra* note 96.

99. 17 U.S.C. § 102(a) (1982).

100. 17 U.S.C. § 106 (1982).

101. Patent protection is not limited to protection of the underlying algorithm for the reason that protection of narrow scope is possible on programs *per se* when proper claim format is used. Most attempts to patent programming are and will be limited to attempts to cover the underlying algorithm. This rationale is sound for the reason that any patent having claims which recite in substance individual program steps or their equivalents would be of such narrow scope as to be practically worth-

C. Protection of Programs Embodied in Object Code

The literal explosion of microprocessor applications has had the effect of increasing investment in programming which is implemented in read only memories (ROMs) and other types of hardware memories used to control microprocessors. Today's technology permits exact duplication of a program which is stored in a ROM or equivalent memory device by the making of identical copies of the original ROM. This facilitates piracy of arcade-type games such as "PAC MAN." An exact copy of a control ROM for a particular arcade game can be used to convert any other arcade game into a duplicate of the particular arcade game without involving great expense or substantial time. The use of copyrights to protect object code embodied in storage media such as ROM's presents a unique challenge to the adaptability of the copyright law to new technologies. Under section 102(b) of the Copyright Act, the "method of operation" is outside the scope of copyright protection. To the extent that it can be effectively argued that object code is primarily a manifestation of a method of operation of a device, copyright protection will be ineffective.

The courts have not reached a consensus on whether a copyright on a source code includes its implementation as object code in a ROM or other similar device; the trend of decisions, however, appears to have shifted toward protection of the object code. The first case on the issue, *Data Cash Systems v. J.S.G.A. Group, Inc.*,[102] held that object code in a ROM was not covered by a copyright on the source code. The court reasoned that "since the ROM is not in a form which one can 'see and read' with the naked eye, it is not a 'copy' within the meaning of the 1909 Act. In its object phase, the ROM, the computer program is a mechanical tool or a machine part but it is not a 'copy' of the source program."[103] The conclusion of the *Data Cash* court conflicts with the section 101 definition of a "copy" and was recently specifically rejected by *Williams Electronics, Inc. v. Artis International, Inc.*.[104] In *Williams Electronics*, object code stored in a ROM was held to be protected by a copyright

---

less. Under the law of patent infringement, a process is not literally infringed if any step is omitted from the accused infringing process.

For a discussion of dual copyright-patent protection, *see infra* text accompanying notes 117-21.

102. 480 F. Supp. 1063 (1979). This case was decided under the old copyright law but the court made it clear in dicta that it believed that the 1976 Copyright Act, if applicable, would yield the same result. *Id.* at 1066 n.4.

103. *Id.* at 1069.

104. 24 P.T.C.J. 387 (1982).

on the source code.[105] This case involved piracy of an arcade game. The defendant argued that a "distinction must be drawn between the 'source code' version of a computer program, which can be copyrighted, and the 'object code' stage which it argued cannot be protected" for the reason that "a computer program stored in a ROM does not satisfy the statutory requirement of being fixed in a material object because a 'copy' must be intelligible to human beings and must be intended as a medium of communication to human beings."[106] The Third Circuit, however, rejected the defendant's argument:

> The answer to defendant's contention is in the words of the statute itself. A 'copy' is defined to include a material object in which a work is fixed 'by any method now known or later developed, and from which the work can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device.' 17 U.S.C. § 101. . . . By this broad language Congress opted for an expansive interpretation of the terms 'fixation' and 'copy' which encompass technological advances such as those represented by the electronic devices in this case. We reject any contention that this broad language should nonetheless be interpreted in a manner which would severely limit the copyright ability of computer programs which Congress clearly intended to protect. We cannot accept defendant's suggestion that would afford an unlimited loophole by which infringement of a computer program is limited to copying of the computer program text but not to duplication of a computer program fixed on a silicon chip. This was also the conclusion reached in *Tandy Corp. v. Personal Micro Computers, Inc.*, . . . albeit in the context of computers rather than video games.[107]

The Court also distinguished *Data Cash* by arguing that the Seventh Circuit affirmance was on other grounds.

In *GCA Corp. v. Chance*,[108] the court sustained the plaintiff's argument that object code stored in a ROM is covered by a copyright on the source code. The court reasoned "[b]ecause the object code is the description of the copyrighted source code, the two are to be treated as one work; therefore copyright of the source code protects the object code as well."[109] Another court, however, has viewed the issue of coverage of object code stored in a ROM by a copyright on the source code to be unsettled. In *Apple Computer, Inc. v. Franklin*

105. *Id.* at 388.

106. *Id.*

107. *Id.*

108. 24 P.T.C.J. 574 (1982).

109. *Id.* at 575.

Computer Corp.,[110] the court refused to grant a preliminary injunction for the reason that "[o]pinion has been divided on how to treat object codes, and on how to treat ROMs generally."[111] The effectiveness of copyright protection will be substantially weakened if the courts do not adopt the rule set forth by the Third Circuit in *Williams Electronics*. A contrary result will substantially enhance piracy, especially in the area of arcade games and similar devices which are easily reverse engineered by copying the control ROM.

The deterrent effect of copyrights on the development of competing software depends to a large extent upon the effort required to produce the program.[112] The protection provided by copyright on a program will be small where the development effort of the coding is a small part of the overall development. Copyright protection will be greater where the development effort of the coding is a large part of the overall development effort. Furthermore, the permissible copying of an algorithm to develop competitive coding can eliminate a large part of a competitor's effort to duplicate a program.

Copyright has several advantages over patent protection. There is no examination of copyrights. The expense of obtaining registration is low. A work is eligible for copyright so long as it is original[113] and contains a minimum of intellectual labor.[114] Courts traditionally will grant preliminary injunctions to stop violation of copyrights. It is almost impossible to obtain an injunction prior to final judgment in a patent infringement action. While it is impossible to quantify the minimum intellectual effort required to make a program copyrightable, programs involving more than a few steps should not be subject to challenge as uncopyrightable subject matter.

VIII. TRADE SECRETS

The protection of programming by trade secrets is effectuated by the application of state law. Variation from state to state in the

110. 25 P.T.C.J. 388-89 (1982). Presumably, the court may have reached a different result if it were aware of the *Williams Electronics* case. *See supra* note 104.

111. 25 P.T.C.J. at 389.

112. *See* Keplinger, *Computer Software—Its Nature and Its Protection*, 30 EMORY L. J. 483 (1981).

113. Under the copyright law, any number of copyrights may be registered on the same work so long as the works to be registered are original to their respective authors. The patent law, on the other hand, only permits the first inventor of an invention to obtain a patent. This is the so-called "novelty" criteria under § 102.

114. *See, e.g.*, Cash Dividend Check Corp. v. Davis, 247 F.2d 458 (9th Cir. 1957); Brown Instrument Co. v. Warner, 161 F.2d 910 (D.C. Cir. 1947); and Taylor Instrument Co. v. Fawley-Brost Co., 139 F.2d 98 (7th Cir. 1943).

enforcement of trade secrets involving programming is, therefore, likely to occur. Twenty-five states and all federal jurisdictions have substantially adopted the definition of trade secrets as contained in section 757 of the *Restatement of Torts*, comment b:

> [a]ny formula, pattern, device or compilation of information which is used in one's business, and which gives him an opportunity to obtain an advantage over competitors who do not know or use it. The subject matter of a trade secret must be . . . secret so that, except by the use of improper means, there would be difficulty in acquiring the information. An exact definition of a trade secret is not possible. Some factors to be considered in determining whether given information is one's trade secret are: (1) the extent to which the information is known outside of his business; (2) the extent to which it is known by employees and others involved in his business; (3) the extent of measures taken by him to guard the secrecy of the information; (4) the value of the information to him and to his competitors; (5) the amount of effort or money expended by him in developing the information; (6) the ease or difficulty with which the information could be properly acquired or duplicated by others.[115]

The key element in maintaining a trade secret is to prevent it from becoming general knowledge. Many factors can cause a trade secret to become unprotectable because of loss of secrecy. Reverse engineering of the trade secret is permissible. Thus, computer programs which are not lawfully protected by patent and copyright are fair game for reverse engineering or copying if they were lawfully obtained. The reverse engineering of an unprotected ROM would be a classic example of a lawfully obtained trade secret. Widespread licensing of trade secrets, under conditions of confidence to a large percentage of potentially interested persons could vitiate the trade secret. This theory of trade secret law was discussed in the Final Report of the National Commission on New Technological Use of Copyrighted Works.[116]

IX. PATENT-COPYRIGHT INTERFACE

There is no conflict between utility patent and copyright protection on programming. In *In re Yardley*,[117] the CCPA held that the USPTO rejection of a design patent application was improper because there was no statutory estoppel consequent from the obtaining of copyrights on the "Spiro Agnew" wristwatch. The CCPA reasoned that Congress has manifested its intention to create two

115. RESTATEMENT OF TORTS § 757 comment b (1939).

116. NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT 17 (July 31, 1978).

117. 493 F.2d 1389 (C.C.P.A. 1974).

distinct statutory rights with potentially overlapping areas of coverage:

> We believe that the 'election of protection' doctrine is in direct conflict with the clear intent of Congress manifested in the two statutory provisions quoted above. The Congress has provided that subject matter of the type involved in this appeal is 'statutory subject matter' under the copyright statute and is 'statutory subject matter' under the design patent statute, but the Congress has not provided that an author-inventor must elect between securing a copyright or securing a design patent. Therefore, we conclude that it would be contrary to the intent of Congress to hold that an author-inventor must elect between the two available modes of securing exclusive rights.[118]

The CCPA's rationale that there is no conflict between design patents and copyrights is equally applicable to the potential overlap between copyrights and utility patents.[119] The same two constitutionally-based[120] statutes are involved in a utility patent-copyright interface as was present in *Yardley*. Moreover, the distinction between the protection conferred by utility patent and copyright is clearly recognized in section 102 of the copyright law where protection of the underlying "idea" is clearly barred. Patent coverage on programming clearly covers the underlying algorithm whereas copyright protection on programming covers only the author's coding. Moreover, subsection (d) section 301 of the copyright law clearly establishes that the copyright law does not affect other federal statutes: "Nothing in this title annuls or limits any rights or remedies under any Federal statute."[121]

X. TRADE SECRET-COPYRIGHT INTERFACE

Section 301 of the copyright law establishes federal preemption of copyright law, and states:

> Preemption with respect to other laws (a) On and after January 1, 1978, all legal or equitable rights that are equivalent to any of the exclusive rights within the general scope of copyright as specified by sections 102 and 103, whether created before or after that date and whether published or unpublished, are governed exclusively by this title. Thereafter, no person is entitled to any such right or equivalent right in any such under the common law or statutes of any State. (b) Nothing in this title annuls or limits any rights or

118. *Id.* at 1394.

119. All of the cases heard by the CCPA and Supreme Court have involved attempts to obtain utility patent coverage.

120. Article I, section 8 of the Constitution provides the basis for both the patent and the copyright laws.

121. 17 U.S.C. § 301(d) (1982).

> remedies under the common law or statutes of any State with respect to—
>
> (1) subject matter that does not come within the subject matter of copyright as specified by sections 102 and 103, including works of authorship not fixed in any tangible medium of expression; or
>
> (2) any cause of action arising from undertakings commenced before January 1, 1978; or
>
> (3) activities violating legal or equitable rights that are not equivalent to any of the exclusive rights within the general scope of copyright as specified by section 106.[122]

The possibility of federal preemption of trade secret law by the copyright law was considered by Congress in the legislative history of the Computer Software Act of 1980,[123] but the courts have yet to rule on this issue.

In *Technicon Medical Information Systems Corp.* v. *Green Bay Packaging, Inc.*,[124] the Seventh Circuit held that documents bearing a copyright notice under the Copyright Act of 1909 are not barred from trade secret protection. The District Court certified the issue on appeal, under 28 U.S.C. section 1292 as "whether the acts of (1) affixing to certain documents a statutory notice of copyright . . . and (2) publishing the document, estops the party who affixed the notice and published the documents from subsequently asserting that such documents have not been generally published as represented by the copyright notice but instead contain subject matter which is trade secret."[125] The Seventh Circuit rejected the defendant's argument that the doctrine of statutory estoppel barred trade secret protection for the reason that no benefit had been conferred by that publication under the terms of the 1909 Act. An *amicus curiae* raised the constitutional argument that the Supremacy Clause of the Constitution precluded state trade secret protection when the

---

122. *Id.* § 301.

123. H.R. REP. NO. 1307, Pt. I, 96th Cong., 2d Sess. 23-24 (1980). The legislative history of the Computer Software Copyright Act of 1980 states:

> Section 12 embodies the recommendations of the Commission on New Technological Uses of Copyrighted Works with respect to clarifying the law of copyright of computer software. During the course of Committee consideration the question was raised as to whether the bill would restrict remedies for protection of computer software under state law, especially unfair competition and trade secret laws. The Committee consulted the Copyright Office for its opinion as to whether section 301 of the 1976 Copyright Act in any way preempted these and other forms of state law protection for computer software. On the basis of this advice and advice of its own counsel the Committee concluded that state remedies for protection of computer software are not limited by this bill.

124. 24 P.T.C.J. 479 (1982).

125. *Id.* at 479-81.

election of copyright protection was made by affixation of a copyright notice. The Seventh Circuit did not find any inherent overlap of legal protection between the federal Copyright Act and state trade secret law, since trade secrets protect the content or ideas of a work while copyright protects the form of the work. The court, in dicta, stated that dual enforcement of rights under copyright and trade secrets could pose a conflict, but the mere assertion of rights did not.[126]

XI. PRACTICAL CONSIDERATIONS

A. PATENTABLE SUBJECT MATTER

The protection of patentable forms of programming should be governed by several practical considerations. The algorithm must satisfy the statutory criteria for patentability: novelty under 35 U.S.C. section 102 and unobviousness under 35 U.S.C. section 103. Thus, a quantum of "inventiveness" is required by the patent law for programming to be patentable. The mere adaption of an old programming algorithm[127] to a slightly different format is not likely to be sufficiently unobvious to be patentable. The projected useful commercial life of the algorithm should be at least two or more years. U.S. Patent and Trademark Office application backlogs make it virtually impossible to obtain a patent in less than two years, and the pendancy of a patent application does not provide any statutory right to exclude practice of an invention before the patent is issued.

The value of the programming should be sufficient to support the expense of patenting. With the new fee legislation that became effective October 1, 1982, which included the provision of patent maintenance fees, the cost of filing an application is likely to be at least $2,000.00 or more considering government and professional service fees, with approximately another $1,450.00 or $2,900.00 in issue and maintenance fees over the seventeen year life of a patent. The cost of prosecuting the application before the U.S. Patent and Trademark office would be additional.

---

126. *Id.*

127. Under 37 C.F.R. § 1.56, the patent applicant has a duty to inform the Patent Examiner of all prior art that a reasonable Examiner would consider "important in deciding whether to allow the application to issue as a patent." The applicant who wishes to patent programming should be particularly careful to inform the Examiner of all prior art of which he knows. The USPTO search facilities do not have a large body of prior art pertaining to programming. Thus, in order to insure USPTO consideration of the most relevant prior art, which is essential to maintenance of the persumption of validity under 35 U.S.C. § 282 in infringement litigation, the applicant should undertake a detailed investigation of all related programming known to it and cite the known programming to the USPTO.

The protection of an algorithm embodied in programs which are intended for widespread distribution by either licensing or outright sale will be difficult or practically impossible under trade secret law. The only effective protection mechanism for the algorithm would be by patent protection, since widespread distribution of the program would destroy any trade secret which resides in the algorithm.

The maximum statutory protection for programs containing patentable algorithms would be by dual patent and copyright protection. The securing of copyright protection, by affixation of notice on the program is accomplished at no expense. Thus, a valuable program algorithm can be protected by patent to preclude the copying of its underlying logic, while its programming can be protected by copyright. If publication is only under conditions of confidence, trade secret protection may also be maintained. In many situations, the high cost of developing software and its substantial commercial value will warrant the seeking of copyright-patent protection as well as attempts to maintain trade secret protection.

A trade secret claim in the algorithm may be maintained up to the time that a patent issues. Under 35 U.S.C. section 122, the Commissioner of Patents maintains the secrecy of all applications "unless necessary to carry out the provisions of an act of Congress or in such special circumstances as may be determined by the Commissioner."[128] Thus, a patent applicant may elect to obtain patent coverage if the scope of patent protection which may be obtained would adequately protect the algorithm or he may abandon the application without any publication where the algorithm would be inadequately protected.

Organizations engaged in the development of programming should review their employment agreements to determine if the employment relation addresses ownership of the development of patentable software. In view of the widespread belief that software is not patentable, it is likely that careful consideration of ownership of patent rights has not been given in many employment agreements where the primary business of the employee is to write software.

B. Unpatentable Subject Matter

The use of copyright protection to protect unpatentable programming in the form of source code or object code is well developed. The low expense of securing copyright protection and the availability of injunctive relief for copyright infringement are distinct advantages of copyright protection. The disadvantage of copy-

128. 35 U.S.C. § 122 (1976).

right protection is that there is no infringement where the underlying algorithm is used to prepare a program intended to accomplish the same result as the copyrighted work. In effect, the copyright owner dedicates the underlying algorithm to the public if no other form of legal protection is available. If the program is to be widely distributed without conditions of confidentiality, there is no effective mechanism for preventing use of the algorithm.

C. Trade Secret Protection

Trade secret protection is widely used to protect programming. The trade secret is usually explicitly defined by contractual terms between the parties which establish conditions of confidence. Covenants not to compete are often used as an additional condition of maintaining trade secret status. Nevertheless, the use of trade secret protection may be impossible where copyright or patent protection is obtained. Any disclosure of the trade secret by unlimited publication of a copyright or in a patent specification will destroy trade secret status to the extent that the trade secret is the same as that contained in a patent specification or the copyright. In situations where the trade secret differs from the subject matter protected by the copyright or patent, however, there is no possible conflict between trade secret, patent and copyright protection.

XII. CONCLUSION

The methods used for protecting programming should be carefully reevaluated in light of the availability of patent protection on many forms of programming. Depending upon the circumstances, patent protection on an algorithm may be a valuable new adjunct to existing methods of protecting programming.