

Summer 1996

## Computer Software Protection in 1996: A Practitioner's Nightmare, 29 J. Marshall L. Rev. 853 (1996)

Jonathan E. Retsky

Follow this and additional works at: <https://repository.law.uic.edu/lawreview>



Part of the [Computer Law Commons](#), [Intellectual Property Law Commons](#), and the [Jurisprudence Commons](#)

---

### Recommended Citation

Jonathan E. Retsky, Computer Software Protection in 1996: A Practitioner's Nightmare, 29 J. Marshall L. Rev. 853 (1996)

<https://repository.law.uic.edu/lawreview/vol29/iss4/3>

This Article is brought to you for free and open access by UIC Law Open Access Repository. It has been accepted for inclusion in UIC Law Review by an authorized administrator of UIC Law Open Access Repository. For more information, please contact [repository@jmls.edu](mailto:repository@jmls.edu).

# COMPUTER SOFTWARE PROTECTION IN 1996: A PRACTITIONER'S NIGHTMARE

JONATHAN E. RETSKY\*

## INTRODUCTION<sup>1</sup>

The evolution of computer software protection law has encountered new bumps on this already rough road of intellectual property. As a result of the unpredictability of computer software protection, practitioners in the field have struggled with their attempts to understand the current state of the law. Recent changes designed to smooth the road have now developed some potholes themselves. However, it is the client — the intellectual property owner — who has suffered the most because of the persistent difficulties with the law. Changes must continue to occur in order to eradicate the problems that both practitioners and clients needlessly face, and pave the way for more certain protection in the future.

The first step towards eliminating the difficulties that currently exist with computer software protection law is understanding the developments that have recently taken place in this continuously changing area. In the field of patent law, the United States Patent and Trademark Office (PTO)<sup>2</sup> has proposed new guidelines that are designed to ease the availability of protection for software and computer-related inventions.<sup>3</sup> Federal Courts have also added to the body of jurisprudence in this area. For

---

\* B.S., Electrical Engineering, 1984, Bradley University, J.D., University of Texas at Austin, 1989. Mr. Retsky practices law in the Chicago office of Brinks, Hofer, Gilson & Lione. His practice includes computer-related IP litigation, drafting and prosecuting software and hardware patent applications, negotiating and drafting licenses and other software agreements, and preparing and filing software copyright applications. Prior to law school, Mr. Retsky was employed by Motorola, Inc., where he helped design the 68030 microprocessor.

1. This article was developed from a speech given by the author in February, 1996, at a conference of intellectual property practitioners. Because the topic of the speech involved timely issues regarding the protection of computer software, this article does not represent nor provide an in-depth substantive analysis of the issues addressed. The article is rather an overview of the current status of protection for computer software, and thus highlights the pertinent issues in this area.

2. The Patent and Trademark Office, a branch of the Commerce Department, is charged with receiving, examining, and issuing U.S. patents, as well as handling re-examinations, re-issues, and interference proceedings.

3. Guidelines for Examination of Computer-Implemented Inventions, 60 Fed. Reg. 28778 (1995) (proposed June 2, 1995).

example, in 1994 the Court of Appeals for the Federal Circuit decided *In re Alappat*.<sup>4</sup> With this *in banc* decision, the court set the high water mark for legal guidance to the patent law practitioner in this historically turbulent area.

In the area of copyright law, the United States Supreme Court recently decided *Lotus Development Corp. v. Borland International Inc.*,<sup>5</sup> where the Court affirmed a holding that a menu command hierarchy for a computer spreadsheet program is not copyrightable because it is a method of operation.<sup>6</sup> Practitioners have questioned whether *Borland* may foretell the waning of the expansive protection that copyright laws currently afford computer software. On the other hand, the Court's decision may have the effect of prolonging some uncertainties in this controversial area of software protection.

This article examines the jurisprudence that, over the last few years, has created the ambiguities that currently exist in computer software protection law. Part I analyzes the protection that patent law affords to computer software and Part II examines copyright protection. Part III discusses some transactional aspects of computer software licensing. Finally, Part IV proposes an alternate and separate scheme of computer software protection, focusing on the technical and commercial aspects of this form of high technology.

## I. PATENT PROTECTION

One way to protect computer software is through the patent laws.<sup>7</sup> Patent protection is awarded to new and useful machines, manufactures, processes, and compositions of matter.<sup>8</sup> The patent law has struggled, however, to fit computer software into one of these categories. Over the years, both the PTO and the courts have fought over the proper level of protection, if any, that the law should afford computer software. Nonetheless, each year American companies spend an enormous amount of time and money on the research and development of this highly lucrative technology.

Section A highlights a number of significant case law deci-

---

4. 33 F.3d 1526 (Fed. Cir. 1994).

5. 116 S. Ct. 804, 804-05 (1996).

6. *Id.* at 804-05; *Lotus Dev. Corp., v. Borland Int'l, Inc.*, 49 F.3d 807, 819 (1st Cir. 1995).

7. See BEIRNE STEDMAN, PATENTS § 1, at 3 (1939). A patent is an agreement between the government and the patentee. *Id.* The patentee agrees to disclose his or her invention so that anyone who is skilled in the area to which the invention appertains may practice and use that invention. *Id.* In return, the government grants the patentee a monopoly over the invention for a period of years. *Id.*

8. 35 U.S.C. § 101 (1994).

sions that have influenced the patentability of computer software. Despite the many decisions in this area, practitioners and clients alike continue to have difficulties understanding the current state of the law. Section B discusses the new PTO guidelines, which were adopted with the purpose of aligning the policies of its office with the recent case law decisions that have affected the industry. Finally, Section C discusses some issues and problems that practitioners in the field are currently facing.

### A. Patent Jurisprudence

As a result of the ambiguity and pervasiveness of the computer software protection jurisprudence, the PTO has proposed new guidelines for its examiners with the idea of conforming its policies with the case law interpretations of the Patent Act.<sup>9</sup> In order to comprehend the PTO's reasoning for its adoption of the new guidelines, it is first essential to understand some significant judicial decisions in this area. The recent landmark decision in this field of software patent protection is *In re Alappat*.<sup>10</sup> An understanding of the case law in this area can be assisted by focusing on those opinions issued prior to *Alappat*, as compared to those opinions rendered after the *Alappat* decision.

Long before *Alappat*, the old Court of Customs and Patent Appeals (CCPA)<sup>11</sup> issued three decisions that outlined a formal test whether the patent laws would afford protection to a computer software invention. The PTO and the Patent Board of Appeals and Interferences<sup>12</sup> followed the two-part test that the cases of *In re Freeman*,<sup>13</sup> *In re Walter*,<sup>14</sup> and *In re Abele*<sup>15</sup> successively and collectively describe.<sup>16</sup> These organizations utilized the Freeman-

---

9. 35 U.S.C. §§ 1-376 (1994).

10. 33 F.3d 1526 (Fed. Cir. 1994).

11. In 1929, Article III of the United States Constitution established the Court of Customs and Patent Appeals, which was the successor to the United States Court of Customs Appeals. BLACK'S LAW DICTIONARY 386 (6th ed. 1990). However, in 1982 the Federal Courts Improvement Act abolished the Court. *Id.* The United States Court of Appeals for the Federal Circuit (CAFC) now handles the matters that the CCPA formerly handled. *Id.*

12. The Patent Board of Appeals Interferences is the first step of appeal within the Patent Office for an applicant whose application an examiner has rejected. The Board handles both appeals from examiner's decisions as well as inter partes interference proceedings.

13. 573 F.2d 1237 (C.C.P.A. 1978).

14. 618 F.2d 758 (C.C.P.A. 1980).

15. 684 F.2d 902 (C.C.P.A. 1982).

16. See *Abele*, 684 F.2d at 905 (holding that the patent laws will not protect a claim that entirely preempts an algorithm, or if it preempts an algorithm but for restricting its use to a specific technological environment); *Walter*, 618 F.2d at 767 (providing that if a mathematical algorithm is implemented in a specific manner such that it defines the structural relationships between the physical elements of a

Walter-Abele test in order to assess the patentability of computer-implemented inventions. The first part of the test focusses on whether an applicant's claim recites a mathematical algorithm.<sup>17</sup> If a claim does recite an algorithm, the second part of the test requires a court to question whether the claim would wholly preempt an entire area of science and technology.<sup>18</sup> According to the PTO, such claims, if allowed, could be read broadly to cover any apparatus capable of implementing an entire body of science or mathematics thus providing far broader protection than the conceived invention.

During the development of this two-part test, the United States Supreme Court decided two cases, *Parker v. Flook*<sup>19</sup> and *Diamond v. Diehr*,<sup>20</sup> both of which impacted the field of computer software protection. These cases aided the CCPA's interpretation of the Patent Act itself, and expounded upon the congressional intent behind the act. In addition, this case law helped the CCPA to promulgate the test upon which the PTO would ultimately focus. *Parker* and *Diamond* also set the groundwork for later decisions<sup>21</sup> that ultimately gave rise to the CAFC's holding in

---

claim, or if the algorithm limits or refines claim steps, then the claim is valid under the Patent Act); *Freeman*, 573 F.2d at 1245 (explaining that if a claim does not recite an algorithm, it clearly cannot preempt an algorithm).

17. *Freeman*, 573 F.2d at 1245 (explaining that a court must first determine whether a claim directly or indirectly recites a mathematical algorithm).

18. *Id.* The PTO wants to avoid awarding a monopoly to an entire field of science or mathematics rather than an utilitarian application of a specific principle.

19. 437 U.S. 584 (1977). *Parker* involved a patent for a mathematical formula for updating alarm limits. *Id.* at 585. The issue of the case involved whether or not the mathematical formula was eligible for patent protection because of the identification of a limited category of post-solution applications of the formula. *Id.* The Court held that mathematical formulas are not eligible for patent protection, even upon the identification of a post-solution application of that formula. *Id.* at 590. The Court explained that the formula in the case at bar merely provided a new method for calculating alarm limit values, and that the formula involved was therefore not a patentable invention. *Id.* at 594-95.

20. 450 U.S. 175 (1981). *Diamond* involved a patent application for an invention of a process for taking raw, uncured synthetic rubber and molding it, in a press, into a cured precision product. *Id.* at 176. The invention involved the use of a mathematical formula which was involved in a process of continuously measuring the inside temperature of a molding press. *Id.* A computer would read the measurements, and repeatedly recalculate the exact amount of time necessary to cure the rubber. *Id.* A device would open the molding press when the rubber was completely cured. *Id.* The *Diamond* court explained that the patent laws do not afford protection to purely mathematical formulas. *Id.* (citing *Gottschalk v. Benson*, 409 U.S. 63, 72 (1972)). However, the court held for the first time that a mathematical formula can be patentable. *Diamond*, 450 U.S. at 192-93. The court explained that if a claim applies a mathematical formula in a process which performs a method that is itself the design of the patent laws to protect, the claim satisfies the statutory requirements of the Patent Act. *Id.* at 192.

21. See, e.g., *In re Donaldson Co., Inc.*, 16 F.3d 1189 (Fed. Cir. 1994); Ar-

*Alappat*. Despite the U.S. Supreme Court's decisions in the area of computer software protection, the PTO focussed its regulations more narrowly than the specific instructions that the courts had provided.

After many years of confusing decision-making, two decisions in this area resulted in an apparent conflict of interpretation between the PTO and the CAFC. The first case, *In re Iwahashi*,<sup>22</sup> was an ex parte proceeding that reached the Federal Circuit from the Patent Office Board of Appeals and Interferences.<sup>23</sup> Not surprisingly, the Board upheld an examiner's rejection that the claims as filed were non-patentable subject matter under § 101.<sup>24</sup> The claims largely included only means-plus-function elements<sup>25</sup> that the examiner read so broadly as to wholly preempt this area of computer technology.<sup>26</sup> In addition to the means-plus-function elements, however, the claim also recited a structural element.<sup>27</sup> This element was a read-only memory (ROM), and was claimed directly rather than in the means-plus-function style of § 112 of the Patent Act.<sup>28</sup> After the Board upheld the examiner's rejection, the applicant appealed the case to the CAFC.<sup>29</sup> The CAFC reversed the Board's decision, holding that the claims recited patentable subject matter.<sup>30</sup> The CAFC criticized the Board as reading the means-plus-function clauses too broadly, and ordered the PTO to reconsider the case consistently with the court's ruling.<sup>31</sup>

The second influential case was the Federal Circuit's decision in *Arrhythmia Research v. Corazonix*.<sup>32</sup> *Arrhythmia* was an inter partes infringement proceeding on a patent that the PTO had is-

---

rhythmia Research Technology, Inc., v. Corazonix Corp., 958 F.2d 1053 (Fed. Cir. 1992); *In re Iwahashi*, 888 F.2d 1370 (Fed. Cir. 1989).

22. 888 F.2d 1370 (Fed. Cir. 1989).

23. *Id.* at 1371.

24. *Id.* In *Iwahashi*, the examiner rejected the patent application explaining that it was a mathematical algorithm and was therefore nonstatutory subject matter under 35 U.S.C. § 101. *Id.*

25. In 1952, Congress amended the Patent Act to allow an applicant to express an element in a claim as a means for performing a specified function. 35 U.S.C. § 112 (1994). The statute reads in pertinent part: An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof. *Id.*

26. *Iwahashi*, 888 F.2d at 1375.

27. *Id.*

28. *Id.*

29. *Id.* at 1371.

30. *Id.* at 1375.

31. *Iwahashi*, 888 F.2d at 1374-75. Mr. Iwahashi and his co-inventors were ultimately awarded U.S. Patent No. 5,007,101.

32. 958 F.2d 1053 (Fed. Cir. 1992).

sued.<sup>33</sup> The United States District Court for the Northern District of Texas found that the issued claims violated § 101 of the Patent Act, and were therefore invalid.<sup>34</sup> On appeal, the CAFC reversed the district court's decision.<sup>35</sup> For once, the CAFC agreed with the Patent Office, and found that the claims recited patentable subject matter.<sup>36</sup>

The *Iwahashi* and *Arrhythmia* cases highlighted a standoff that had developed between the CAFC and the PTO regarding the interpretation of means-plus-function elements in software inventions.<sup>37</sup> This was a standoff that the CAFC needed to resolve. For some time, the PTO taunted the CAFC's authority by ignoring CAFC decisions on this issue, and challenging CAFC authority to instruct the PTO on the examination of patent applications. The PTO's position, which it asserted before the CAFC, relied on prior CCPA jurisprudence to interpret patent claims. According to the PTO, only an *in banc* decision from the CAFC could overrule the CCPA cases they chose to follow.

Two 1994 decisions finally set the matter straight. In both cases, the CAFC met *in banc* and finally addressed the issue of what constitutes patentable subject matter under § 101 of the statute for computer software inventions. In *In re Donaldson*,<sup>38</sup> the CAFC formally admonished the PTO for interpreting means-plus-function elements too broadly and ignoring the plain meaning of the statute.<sup>39</sup> The court instructed the Patent Office to read means-plus-function elements as set forth in the statute to cover the "structure, material or acts described in the specification and equivalents thereof."<sup>40</sup> A broader reading would contradict the congressional intent behind this aspect of the Patent Act.<sup>41</sup> The *Donaldson* decision also established the necessary basis for the CAFC's reasoning in the *Alappat* decision.

In *Alappat*, a plurality of six judges, with three dissenters and three abstainers, finally ruled *in banc* on what constitutes patentable subject matter under § 101 for computer software in-

---

33. *Id.* at 1055.

34. *Id.* at 1054 (explaining that the district court had held that the claim consisted of a mathematical algorithm and was therefore nonstatutory subject matter).

35. *Id.* at 1054.

36. *Id.*

37. The PTO interpreted means plus function elements to cover everything conceivable to perform the stated function. The CAFC interpreted means plus function elements to cover only the corresponding structure, material or acts described in the patent specification, as mandated in 35 U.S.C. § 112 (1994).

38. 16 F.3d 1189 (Fed. Cir. 1994).

39. *Id.* at 1193.

40. *Id.* (quoting 35 U.S.C. § 112 (1984)).

41. *Donaldson*, 16 F.3d at 1194 (explaining that Congress intended for paragraph six of § 112 to apply to every interpretation of a means plus function claim and not merely to the interpretation of post-issuance claims).

ventions.<sup>42</sup> Ultimately, the court focused on the legislative history behind § 101 and concluded that Congress intended § 101 to cover "everything under the sun that is made by man"<sup>43</sup> with only three exceptions.<sup>44</sup> The three exceptions were established by the Supreme Court in *Diamond v. Diehr*.<sup>45</sup> These exceptions are laws of nature, natural phenomenon, and abstract ideas.<sup>46</sup> Other than these three specific areas, therefore, an inventor is entitled to protection of a computer software invention provided, of course, that the invention is both novel and non-obvious.<sup>47</sup>

One can argue that the *Alappat* decision is the most settling or definitive decision to date in computer-implemented invention jurisprudence. With the *Alappat* decision, the court provided a clear and understandable standard to guide the practitioner in protecting software inventions. The *Alappat* decision dispensed with the vagaries of interpretation of the two-part Freeman-Walter-Abele test and replaced them with a clear boundary. No tests are necessary any longer. Other than the basic laws of science and nature, claims covering methods or processes that employ computer technology are patentable subject matter.

Unfortunately, there may be trouble in this short-lived paradise. Some of the decisions since *Alappat* may have re-muddied the waters regarding the patentability of computer software. For example, shortly after *Alappat*, a three judge panel of the CAFC decided *In re Trovato*.<sup>48</sup> Interestingly, the panel consisted of the dissenting judges in *Alappat*. It was apparent to the legal community that with the outcome and reasoning of *Trovato*, the *Alappat* dissenters may have had their revenge.

In a strongly worded opinion, the panel ruled that the subject claims were non-patentable subject matter.<sup>49</sup> In support of its decision, the panel emphasized the absence of any hardware description provided in the specification.<sup>50</sup> The specification focussed solely on software and flowcharts, which, according to the panel, was not enough to gain patentability in and of itself.<sup>51</sup> The court felt that the claimed invention was closer to an abstract idea than a practical application, and rejected the claims.<sup>52</sup> Some of

---

42. *Alappat*, 33 F.3d at 1541-45.

43. *Id.* at 1542 (quoting *Diamond v. Chakrabarty*, 447 U.S. 303, 309 (1980)).

44. *Alappat*, 33 F.3d at 1541-45.

45. 450 U.S. 175 (1981).

46. *Id.* at 185.

47. *Id.* at 191. See 35 U.S.C. §§ 102-03 (1994).

48. 42 F.3d 1376 (Fed. Cir. 1994), *vacated and remanded*, 60 F.3d 807, 807 (Fed. Cir. 1995).

49. *Id.* at 1383.

50. *Id.* at 1380.

51. *Id.*

52. *Id.* at 1381.



the arguments behind its reasoning smacked of the philosophy of the dissent in *Alappat*.

In an extraordinary and illuminating turn of events, however, the CAFC reversed its *Trovato* holding.<sup>53</sup> Without warning, the *Trovato* decision was vacated and withdrawn only months after publication by the very panel that rendered the decision.<sup>54</sup> The reversal begged the question of the motivation for this unusual development. One can only assume that the panel of dissenters ultimately bowed to internal pressure from their majority brethren to abide by the ruling of the *Alappat* decision and not to cast doubt upon it.

Several other post-*Alappat* cases may have added to the confusion, and also set the stage for the Patent Office to adopt its proposed guidelines. These cases include *In re Warmerdam*,<sup>55</sup> and *In re Lowrey*.<sup>56</sup> In the respective decisions, the CAFC addressed whether data structures, a technique which programmers commonly use for the organization of data,<sup>57</sup> are patentable subject matter.<sup>58</sup> In *Warmerdam*, the CAFC rejected the method claims applied for and held that an applicant who had submitted a claim for a data structure was attempting to patent an abstract idea.<sup>59</sup> The court explained that data structures, per se, were not patentable.<sup>60</sup> In *Lowrey*, a similar case posing a similar question, the CAFC held that claims reciting a data structure were in fact patentable.<sup>61</sup> It is difficult to reconcile the reasoning behind and within these apparently conflicting opinions, and even more difficult to apply their teachings in the practice. As a result of these recent opinions, confusion may have again crept into the jurisprudence of patentable subject matter for software-related inventions.

From the high-water mark that the *Alappat* decision presented in 1994, subsequent decisions may have caused the level of protection available to software inventions under the patent laws to recede. These cases are not entirely clear or even consistent, and they are not very instructional to the practitioner. In order to

---

53. *In re Trovato*, 60 F.3d 807, 807 (Fed. Cir. 1995).

54. *Id.*

55. 33 F.3d 1345 (Fed. Cir. 1994).

56. 32 F.3d 1579 (Fed. Cir. 1994).

57. One often cited reference defines a data structure as "[a] physical or logical relationship among data elements, designed to support specific data manipulation functions." *Warmerdam*, 33 F.3d at 1362 (citing THE NEW IEEE STANDARD DICTIONARY OF ELECTRICAL AND ELECTRONICS TERMS 308 (5th ed. 1993)).

58. *Warmerdam*, 33 F.3d at 1360; *Lowrey*, 32 F.3d at 1583.

59. *Warmerdam*, 33 F.3d at 1360. Interestingly, however, the court upheld a claim that merely recited the data structure as being stored in a memory. *Id.*

60. *Id.* at 1362 (explaining that the claim lacked the requisite statutory subject matter).

61. *Lowrey*, 32 F.3d at 1584. All of the claims in *Lowrey* recited a data structure being stored in a memory. *Id.*

eradicate the current ambiguities in the field, the CAFC must provide clearer guidance in this already complex field of technology. Nonetheless, despite these recent setbacks, the *Alappat* decision still shines as the guiding light to the practitioner that broad protection for computer software under the patent laws is available.

### B. Patent Office Practice

In order to align the policies of its office with case law decisions that, over the past fifteen to twenty years, have affected the patentability of computer software, the PTO recently proposed and adopted a number of new guidelines. The Patent Office's first proposed new guidelines for the handling of software related patent applications in June of 1995.<sup>62</sup> The PTO published these guidelines in the form of a request for comments on proposed rule making under the Administrative Procedure Act.<sup>63</sup> In October of 1995, the PTO published a legal analysis that explains and supports its proposed guidelines.<sup>64</sup> This published legal analysis confirms the PTO's intention to conform the policies of its office with the case law that has long since developed in this volatile area.<sup>65</sup>

The case that apparently culminated in the PTO's decision to propose its guidelines was *In re Beauregard*.<sup>66</sup> In *Beauregard*, the PTO agreed to withdraw its appeal to the CAFC from an adverse Board decision that in fact was based on recent CAFC decisions.<sup>67</sup> The PTO agreed that claims directed towards computer software stored on a physical medium,<sup>68</sup> may consist of patentable subject matter if the claims expressly recite an article of

---

62. Guidelines for Examination of Computer-Implemented Inventions, 60 Fed. Reg. 28778, 28778-780 (1995) (proposed June 2, 1995). The proposed guidelines were drafted to assist PTO employees in their examination of patent applications for computer-implemented inventions. *Id.* The Patent Office designed its guidelines to embrace and reflect recent case law decisions from the CAFC that affected patent protection of computer software inventions. *Id.*

63. Request for Comments on Proposed Examination Guidelines for Computer Implemented Inventions, 60 Fed. Reg. 28778 (1995). Hearings were held in the summer of 1995 in San Jose, California and Arlington, Virginia on the proposed guidelines.

64. Request for Comments on Proposed Examination Guidelines for Computer Implemented Inventions 60 Fed. Reg. 52655, 52655 (1995) (providing that interested persons could obtain a legal analysis of the proposed guidelines on the World Wide Web at the following address: <<http://www.uspto.gov/web/software>>.

65. See Examination Guidelines for Computer-Related Inventions, 61 Fed. Reg. 7478, 7479 (1996) (explaining that the PTO intended for its guidelines to be consistent with case law decisions affecting computer-related inventions).

66. 53 F.3d 1583 (Fed. Cir. 1995).

67. *Id.*

68. An example of a physical medium would be an integrated circuit, a magnetic memory or a floppy diskette.

manufacture.<sup>69</sup> The PTO issued its new proposed guidelines shortly after the *Beauregard* decision. These guidelines thus appear as a formal reconciliation by the PTO to finally abide by the jurisprudence of the CAFC in this area.

Surprisingly, and shortly after the PTO published its guidelines and legal analysis, the Patent Office quietly began to suspend the prosecution of software applications that were subject to the utility rejections of 35 U.S.C. § 101.<sup>70</sup> An obscure practice rule allows the PTO to suspend the prosecution of an application for up to six months at a time.<sup>71</sup> The PTO began issuing notices to applicants informing them of the suspension, and instructing the applicant to contact the Patent Office in six months to check on the status of the suspension.<sup>72</sup>

It quickly became apparent that the PTO was struggling with the implementation of its proposed guidelines. It was rumored that Bruce Lehman, the Commissioner of the PTO,<sup>73</sup> had hired or was attempting to hire, new examiners in the software arts group in order to better handle the vast number of software applications being filed.<sup>74</sup> The hiring and training of these new examiners may have proven to be too difficult a task under the proposed guidelines.

Unexpectedly, the PTO then published a new and revised version of the guidelines that the office had originally proposed.<sup>75</sup>

---

69. *Beauregard*, 53 F.3d at 1583.

70. A utility rejection under 35 U.S.C. § 101 is the legal basis for a PTO rejection of a patent application that consists of a mathematical algorithm.

71. See UNITED STATES DEPARTMENT OF COMMERCE - PATENT AND TRADEMARK OFFICE, MANUAL OF PATENT EXAMINING PROCEDURE § 709 (5th ed. 1993) [hereinafter MPEP].

72. An example of the language of such a notice is as follows:

In conformity with M.P.E.P. § 709, awaiting final promulgation of the *Proposed Examination Guidelines for Computer-Implemented Inventions*, U.S. Patent and Trademark Office, Official Gazette, 1175 OG 86-88, June 27, 1995, Ex Parte prosecution of this application has been *SUSPENDED* under 37 C.F.R. § 1.103(b) FOR A PERIOD OF SIX MONTHS FROM THE DATE OF THIS LETTER.

Upon expiration of the period of suspension, applicant should make an inquiry as to the status of the application.

Letter from Jacques H. Louis-Jacques, Patent Examiner, United States Department of Commerce - Patent and Trademark Office (Sept. 25, 1995) (on file with *The John Marshall Law Review*).

73. Bruce A. Lehman is an Assistant Secretary of Commerce, as well as the Commissioner of Patents and Trademarks.

74. See 35 U.S.C. § 131 (1952) (providing that the commissioner shall issue a patent if it appears, upon an examiner's review of an alleged new invention, that an applicant is entitled to a patent under the law). See 35 U.S.C. §§ 101-103 for the statutory conditions precedent to the grant of a patent.

75. Examination Guidelines for Computer-Related Inventions, 61 Fed. Reg.

This new publication provides greater detail regarding the background jurisprudence than did the first proposal. Moreover, the new guidelines focus more on the examination of applications — for example, what steps an examiner should take to guarantee a proper examination.<sup>76</sup> The new guidelines were initially published for discussion purposes only.<sup>77</sup> However, the office finalized these guidelines for official use on February 28, 1996.<sup>78</sup> The new guidelines explicitly replace the originally proposed guidelines and the supporting legal analysis paper.<sup>79</sup>

A review of the final guidelines reveals that they are more comprehensive and practicable than the Patent Office's initial proposal. The final guidelines present a clear roadmap, for both the examiner and the practitioner, of the potential rejections for software patent applications and how to avoid such rejections. Furthermore, the new guidelines better demonstrate that after many years of disagreement, the Patent Office has capitulated to Federal Circuit jurisprudence under § 101 of the Patent Act. Once the PTO resolves its internal practice and training problems, the new guidelines should serve to promote the more efficient issuance of valid patents for computer software inventions.

### C. Practice Issues

For a practitioner to successfully direct his or her client in an area that will eventually provide the necessary blanket of protection, there are many important issues that a practitioner must first consider. One issue that commonly arises in drafting and prosecuting patent applications for computer software is whether an inventor must file a copy of the program source code along with the application. A related question is whether flowcharts illustrating the steps taken by the program are also necessary. Over the years, several cases have addressed these issues.<sup>80</sup> The

---

7478 (1996).

76. *Id.* at 7478-87.

77. A legend indicated that the new guidelines were "under review in the Executive Branch." U.S. Patent and Trademark Office, Examination Guidelines for Computer-Related Inventions <<http://www.uspto.gov>> (last modified Mar. 6, 1996).

78. Examination Guidelines for Computer-Related Inventions, 61 Fed. Reg. 7478, 7478 (1996).

79. *Id.* at 7487 n.1.

80. *See, e.g.*, Northern Telecom, Inc. v. Datapoint Corp., 908 F.2d 931, 940 (Fed. Cir.), *cert. denied*, 498 U.S. 920 (1990) (holding that the amount of disclosure for computer programs varies according to the nature of the invention, the role of the program in carrying out the invention, and the complexity of the program, all viewed from the standpoint of a skilled programmer); White Consolidated Indus. v. Vega Serro-Control, Inc., 713 F.2d 788, 791 (Fed. Cir. 1983) (providing that a valid patent application must fully and clearly describe an invention such that a person

short answer is that, in order to be safe, an applicant should file both the program source code and flowcharts. The more difficult question is whether source code and flowcharts are actually required with every patent application.

The applicable cases place a heavy burden on an applicant to file flowcharts along with the application.<sup>81</sup> Computer programmers, however, have long since stopped using flowcharts in the course of writing computer software programs. Flowcharting the algorithm behind a program was a technique that programmers used in the early days of programming as a precursor step to actually writing the program. It is a time consuming technique that professors may still use to teach software programming. Today, however, practitioners in the industry rarely employ the use of flowcharts, particularly in view of the competitive and fast-paced nature of the computer business. Therefore, it is difficult to ask a programmer, after he or she has written a program, to then create a flowchart solely for the purpose of filing a patent application.

The courts should decide whether flowcharts are an essential element of a patent application, or whether an alternative overview of the algorithm behind a program would suffice. For example, common outlining techniques, which an inventor or his agent can easily prepare at the time of filing an application, would prove a suitable and sufficient alternative to flowcharts. Such an outline would satisfy the disclosure obligation, i.e., to provide an enabling description of the manner and process for making and using the invention, while at the same time emphasize the substance of the disclosure rather than its form. Practitioners can then give flowcharting the just burial that it deserves as an obsolete relic in the history of software and computer evolution.

Another problem with flowcharts is that they do not directly apply to the programming techniques that the industry has recently embraced. Flowcharts work well where structured programming techniques are used.<sup>82</sup> Nowadays, however, most programs are written using "object-oriented" programming techniques. Inventors tend to use these techniques in event-driven systems or in

---

who is skilled in the art will be able to recreate and use the inventor's idea without undue experimentation). The *White* court explained that an inventor's disclosure may be sufficient even if a person skilled in the art must experiment in order to use the invention. *Id.* However, the necessary amount of experimentation cannot be unreasonable. *Id.* See also *In re Sherwood*, 613 F.2d 809, 816-19 (C.C.P.A. 1980) (holding that an applicant's failure to include a copy of a working computer program with his patent application was not fatal to his claim), *cert. denied*, *Diamond v. Sherwood*, 450 U.S. 994 (1981).

81. See, e.g., *Northern*, 908 F.2d at 940; *White*, 713 F.2d at 791.

82. Structured programming comprises a series of computer instructions that typically flow sequentially during execution from a start point to an endpoint.

client-server environments, which programmers are currently developing for the Internet or other multiple computer networks. Object-oriented programming is not inherently linear, so a problem exists in trying to describe or represent a sequential flow of the underlying algorithm of this type of program. In fact, programmers purposely write these programs to be non-linear, and a flowchart does not readily describe the functioning of this software. A combination functional block diagram superimposed with events that cause a response from the program has been useful in the practice to satisfy this obligation of disclosure. An hierarchical representation of the relationship among objects may also help satisfy the enablement requirement of the statute.<sup>83</sup>

## II. COPYRIGHT PROTECTION

The counterpart to patent protection for computer software is in the area of copyright law.<sup>84</sup> Computer software can be simultaneously protected both under the patent laws and under the copyright laws. Indeed, both forms of protection complement each other and serve to protect different aspects of computer software. Over the last few years, significant developments in the area of copyright jurisprudence have continued to define what constitutes copyrightable subject matter.

In the field of copyright law, software is unique because, unlike a book or a play, it is functional by nature. That is, computer software performs a task or it generates an output. This fact gives rise to an issue regarding the extent of an author's rights in the functional or non-literal aspects of a work of authorship. An author can obtain protection for the literal aspects of a computer program under traditional copyright law.<sup>85</sup> However, practitioners are uncertain as to the amount of protection that copyright law provides, if any, for a program's functional aspects. The first part of this section identifies case law decisions that have affected and defined the extent to which the copyright laws protect com-

---

83. See 35 U.S.C. § 112 (1994).

84. Pursuant to 17 U.S.C.A. § 102(a): Copyright protection subsists, in accordance with this title, in original works of authorship fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device. 17 U.S.C.A. § 102(a) (1995). The works of authorship this statute affords protection to includes literary works. *Id.* Courts classify computer programs as "literary works," and as a result copyright law extends protection thereto. *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1234 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).

85. The literal aspects of a computer software invention, as opposed to the underlying idea thereof, would include a printout or a screen display. The copyright laws extend only to protect an inventor's expression of an idea, and not to the idea itself. *Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 703 (2d Cir. 1992).

puter software. The second part of this section discusses the practices of the Copyright Office, and how its policies have created problems for both attorneys and clients in this rapidly evolving field.

### A. Copyright Jurisprudence

There are several decisions that have addressed the copyrightability of computer software over the years. The first important case to define a test for the copyrightability of computer software was *Whelan v. Jaslow*.<sup>86</sup> The court in *Whelan* held that the copyright laws extend protection not only to the literal text of a program, but also to the structure, sequence and organization of a computer program.<sup>87</sup> Under the copyright law, however, ideas are not protected.<sup>88</sup> In order for a court to determine the extent of this non-literal protection it must look for the idea behind each program and exclude that idea from protection.<sup>89</sup> Everything that remains in the program, including both the literal text and the non-literal structure, sequence and organization of the program, constitutes protectable expression.<sup>90</sup>

Despite its enticing simplicity, the *Whelan* court's decision has proven to be an unrealistic and an impractical standard for the complex computer programs that this growing industry is currently developing. The Second Circuit's decision in *Computer Associates International, Inc. v. Altai, Inc.*,<sup>91</sup> jettisoned the *Whelan* test as overly simplistic, concluding that the court's approach incorrectly presumed that there was only one idea behind each computer program.<sup>92</sup> Under *Whelan*, everything else in a

---

86. 797 F.2d 1222 (3d Cir. 1986), cert. denied, 479 U.S. 1031 (1987).

87. *Whelan*, 797 F.2d at 1248. The *Whelan* court explained that there are substantial costs involved in developing a computer program's structure. *Id.* at 1237. In holding that copyright law protects more than the literal code of a computer program, the court has afforded protection to an author's most valuable creation. *Id.*

88. See 17 U.S.C.A. § 102(b) (1995), which provides that: In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work. *Id.*

89. *Whelan*, 797 F.2d at 1236. A court must distinguish between expressions and ideas, as copyright law affords protection to the former but not the latter. *Id.* at 1235-36. In order to make this decision, a court must determine the goal that the author wrote the work to accomplish. *Id.* at 1236. With respect to utilitarian works, the function and purpose of an invention is an uncopyrightable idea. *Id.* However, copyright law does protect an expression of an idea. *Id.* An expression of an idea, therefore, is everything that is not required for the function or purpose of the work. *Id.*

90. *Id.* at 1236.

91. 982 F.2d 693 (2d Cir. 1992).

92. *Id.* at 705. It was the reality of a computer program's structural design, and

computer program, except for that one very basic top-level idea, was protectable expression.<sup>93</sup> In reality, however, computer programs are no longer that simple. There are typically many ideas expressed in a computer program, including an overall idea and many sub-ideas.<sup>94</sup> However, these underlying sub-ideas may not constitute copyrightable subject matter under classic copyright principles.<sup>95</sup> According to *Altai*, a court must also address each sub-idea in its search to find copyrightable expression.<sup>96</sup>

The Second Circuit developed the *Altai* Abstraction-Filtration-Comparison test in order to analyze whether copyrightable subject matter exists for computer programs.<sup>97</sup> In the Abstraction step a court must examine the program to identify all of the various ideas and sub-ideas that the program solves or performs.<sup>98</sup> It may benefit a court, because of the difficulties in dissecting computer programs, to enlist the aid of a programming expert to perform this step. Once a court has identified the various ideas and sub-ideas of a program, the next step is the Filtration process.<sup>99</sup> The Filtration process employs traditional copyright notions, and

---

not the program's purpose, which was the focus of the *Altai* court's criticism. The *Altai* court explained that the ultimate purpose or function of a computer program is achieved through the interaction of the program's subordinate parts. *Id.* Each subordinate part consists of a program, and therefore, an idea. *Id.* Accordingly, because a computer program consists of many subordinate parts, it also consists of many ideas. *Id.*

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

94. *Altai*, 982 F.2d at 705.

95. *Id.* at 708. The *Altai* court, for example, explained that the doctrine of merger provides that if there is only one method available for an author to express an idea, the idea and the expression thereof are indistinguishable. *Id.* at 707-08. In such a case, the expression has merged with the idea, and consequently copyright law does not protect the expression. *Id.* at 708. A court must ask a threshold question in order to decide if the merger doctrine will thwart the extension of copyright protection to a subordinate part of a computer program. *Id.* The court must query as to whether or not a particular subordinate part of a computer program is essential to its implementation. *Id.* If a court answers this question in the affirmative, then the expression that the subordinate part represents has merged with the idea. *Id.* As a result, copyright law will not extend to the expression. *Id.* See 17 U.S.C.A. § 102 (b) (1995) (providing that copyright protection does not extend to ideas). Other classical copyright tests also apply, such as *scenes a faire*, etc. For an explanation of *scenes a faire*, see *infra* note 101.

96. *Altai*, 982 F.2d at 706. Initially, a court must divide a program into its subordinate parts. *Id.* Thereafter, a court must analyze each subordinate part, searching for elements that an author has removed from the public domain, as well as incorporated ideas and the expressions incidental to those ideas. *Id.* A court can then decide which parts of the program copyright laws will protect. *Id.*

97. *Id.* at 706.

98. *Id.* at 707. An important aspect of the abstraction test is that a computer program may contain a multitude of ideas and expressions of ideas. *Id.*

99. *Id.* at 707. The design of the filtration process is to define the extent of an author's copyright. *Id.*



applies those notions to the various ideas and sub-ideas of the program to determine what constitutes copyrightable subject matter.<sup>100</sup> A court can apply theories like *scenes a faire* and the merger doctrine to determine what expression, if any, is original, creative and subject to copyright protection.<sup>101</sup> After a court has filtered out the non-protectable elements of a program, the court must then compare the protected expression with the accused program and determine if it has copied the copyrightable portions.<sup>102</sup> The Abstraction-Filtration-Comparison test is now the most accepted standard regarding whether non-literal elements in a computer program constitute protectable expression under the copyright laws.<sup>103</sup>

A recent case which significantly altered the playing field in the copyright area is a 1996 United States Supreme Court opinion, *Lotus Development Corp. v. Borland International, Inc.*<sup>104</sup> The dispute in *Borland* revolved around the "menu command hierarchy" of the Lotus 1-2-3 spreadsheet program.<sup>105</sup> The menu

---

100. *Id.* at 707.

101. *Altai*, 982 F.2d at 707-10. The *scenes a faire* doctrine recognizes that an author cannot write about a specific period in history without using certain commonly used literary representations. *Id.* at 709 (citing *Hoehling v. Universal City Studios, Inc.*, 618 F.2d 972, 979 (2d Cir.), *cert. denied*, 449 U.S. 841 (1980) (explaining that in a copyright infringement action involving works about the Hindenburg tragedy, the author's use of certain traditional aspects of life in Nazi Germany — for example representations of a tavern, or the Nazi greeting "Heil Hitler" — did not rise to a level of infringement)). The *Altai* court explained that a court can also apply the *scenes a faire* doctrine when analyzing computer programs. 982 F.2d at 709 (citing *Data East USA, Inc., v. Epyx, Inc.*, 862 F.2d 204, 208 (9th Cir. 1988) (holding that a court can apply the *scenes a faire* doctrine, which provides that copyright protection does not extend to standard representations of the expression of an idea, when analyzing computer video games)). See *supra* note 95 for an explanation of the doctrine of merger.

102. *Altai*, 982 F.2d at 710.

103. *Apple Computer, Inc., v. Microsoft Corp.*, 35 F.3d 1435, 1445 (9th Cir. 1994), *cert. denied*, 115 S. Ct. 1176 (1995) (recognizing in dictum that other courts have accepted and utilized the abstraction-filtration-comparison test); *Engineering Dynamics, Inc., v. Structural Software, Inc.*, 26 F.3d 1335, 1342 (5th Cir. 1994) (providing that courts should employ the three part abstraction-filtration-comparison test when deciding whether copyright protection should extend to a computer program); *Gates Rubber Co. v. Bando Chem. Indus.*, 9 F.3d 823 (10th Cir. 1993) (adopting, in substantial part, the abstraction-filtration-comparison test). The Fourth Circuit had an opportunity to adopt the test; however, the court declined, explaining that the facts of the case were not conducive to that type of conclusion. *Avtec Sys., Inc. v. Peiffer*, 21 F.3d 568, 571 (4th Cir. 1994). The Third Circuit apparently still follows the *Whelan* test. See *Maclean Assoc., Inc., v. Wm. M. Mercer-Meidinger-Hansen, Inc.*, 952 F.2d 769, 777 n.6 (3d Cir. 1991).

104. 116 S.Ct. 804, 804-05 (1996).

105. *Lotus Dev. Corp., v. Borland Int'l, Inc.*, 49 F.3d 807, 815 (5th Cir. 1995). In *Borland*, Borland International Incorporated had designed a spreadsheet program which purposely accepted the same command sequences of the Lotus 1-2-3 spreadsheet program. *Id.* at 810.

command hierarchy comprises a sequence of commands shown textually to the user in such a manner so that the user can type a sequence of keys that represent the first letter of each one of the desired commands.<sup>106</sup> To get the Lotus spreadsheet program to perform specific functions, the user simply needs to type the first letter of each command in the proper sequence.

The District Court of Massachusetts erroneously concluded that Lotus' menu command hierarchy was protectable expression under the copyright laws.<sup>107</sup> However, in the First Circuit, the Appellate Court ruled that the Lotus 1-2-3 spreadsheet program's menu command hierarchy was not copyrightable subject matter.<sup>108</sup> The copyright statute expressly provides that its blanket of protection does not extend to methods of operation.<sup>109</sup> The First Circuit concluded that the Lotus 1-2-3 menu command hierarchy was in fact a method of operation, and therefore not copyrightable subject matter.<sup>110</sup>

The Supreme Court then reviewed the First Circuit's decision.<sup>111</sup> With the entire industry holding its breath, the Supreme Court rendered an equally divided per curiam decision affirming the judgment of the appellate court.<sup>112</sup> Unfortunately, the Supreme Court gave no explanation for its decision, and as a result

---

106. The Lotus 1-2-3 spreadsheet has its computer commands organized in a command hierarchy, or menu tree. SARAH E. HUTCHINSON ET AL., LOTUS 1-2-3 VERSION 2.2 20 (1993). The program organizes subsections into a menu of choices under a main menu of command headings. *Id.* The difference between Lotus' user interface and the interfaces being developed today is that no graphics were provided in the menu command hierarchy. Nowadays, more creativity goes into the graphics of a user interface. It may be more difficult to try to compare graphic user interfaces now that they include more color and characters, in addition to the text, to determine whether there is protectable expression. There likely is protectable expression, however, at that level of authorship and originality.

107. *Lotus Dev. Corp. v. Borland Int'l., Inc.*, 831 F. Supp. 223 (D. Mass. 1993) (rejecting the argument that the menu command hierarchy was a method or a system).

108. *Lotus Dev. Corp., v. Borland Int'l., Inc.*, 49 F.3d 807, 819 (1st Cir. 1995).

109. *Borland*, 49 F.3d at 815. Borland conceded that it had copied the Lotus 1-2-3 menu tree. *Id.* at 812. However, Borland argued that the menu tree was not copyrightable pursuant to 17 U.S.C. § 102(b). *Id.* The court agreed with Borland's assertion, holding that copyright protection does not extend to the Lotus menu command hierarchy because it is a method of operation. *Id.* at 815. The court explained that the phrase "method of operation" as used in § 102(b) refers to what means a person uses to operate something; for example a food processor, a car, or a computer. *Id.* Copyright protection simply does not extend to a written description of how to operate something. *Id.* The court further explained that the user operates and controls the Lotus 1-2-3 spreadsheet with the menu command hierarchy. *Id.* Therefore, the Lotus 1-2-3 menu tree is a "method of operation," and uncopyrightable. *Id.* at 818.

110. *Id.* at 815.

111. *Borland*, 116 S. Ct. at 804-05.

112. *Id.* Justice Stevens took no part in the decision. *Id.*

failed to provide a uniform federal precedent to govern this highly controversial issue.<sup>113</sup> Practitioners can only speculate that tough policy questions and industry pressures polarized the justices on this complex issue that is of extreme importance to the computer industry. The per curiam decision, of course, neglects to identify how each justice voted, thereby preventing any measure of predictability as to how the Supreme Court may decide this issue in the future. Undoubtedly, this issue will visit the Supreme Court again because conflicts exist among the regional circuits regarding what constitutes copyrightable subject matter for computer software. At least for the time being, however, it appears that the First Circuit decision is the law of the land.<sup>114</sup>

The *Borland* decision may foretell that the pendulum of copyright law is swinging away from broad protection for computer software and back towards more moderate, limited protection. For the past ten years, case law decisions have allowed the copyright laws to afford expansive protection to the non-literal aspects of computer software. Today, however, courts may be finding that the pendulum has swung too far. It is possible that copyright protection for computer software is contracting to cover only the literal aspects of a computer program. As a result, other areas of the law, for example patent protection, must expand to cover and protect the non-literal, conceptual ideas behind a computer program. The case law regarding software copyright protection, therefore, still remains unsettled.

### *B. Copyright Office Practice*

Beyond understanding the case law decisions in this area, there are numerous matters of importance that copyright practitioners must consider in order to provide the appropriate amount of protection for computer software programs. The main issue is the nature and extent of the deposit material that the Copyright Office requires. In addition to registering an author's copyright for original works of authorship, the Copyright Office performs a collection function for the Library of Congress. A deposit of a copy of

---

113. *Id.*

114. Practitioners may still question whether the district court's decisions in *Borland* are still good law despite the ultimate outcome of the case on appeal. See *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 831 F. Supp. 223 (D. Mass. 1993); *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 779 F. Supp. (D. Mass. 1992). These underlying decisions were helpful in creating an understanding of what is required to develop a "clean room" design of non-infringing computer software. These cases, to the extent that they are not tied to the ultimate question of copyrightability, are still instructive and presumably valid. However, with respect to those portions of the decisions that assume that the menu command hierarchy is copyrightable subject matter, the Supreme Court and the First Circuit decisions control. *Borland*, 49 F.3d at 819, *aff'd*, 116 S.Ct. at 804-05.

the underlying work must therefore accompany an application for copyright registration. Unlike the Patent Office,<sup>115</sup> however, the Copyright Office requires that an applicant's deposit of identifying material for computer software include a paper copy of the program itself.<sup>116</sup> Practitioners in the field find that the deposit of a paper printout of the program cumbersome, difficult, and inefficient.

A desirable alternative would be for the Copyright Office to adopt some form of machine readable deposit copy rules. Machine readable magnetic media is a still more preferable alternative to microfiche because magnetic media is the medium and format that software naturally takes. With the aid of a personal computer, practitioners could readily collect and transmit deposit copies, and the Copyright Office could more easily store, retrieve, and reproduce this material. If the Copyright Office were to adopt such a policy, making and providing deposit copies would be easier, cheaper and less cumbersome for all those involved in registering a copyright.

The rules of the Copyright Office also recognize an applicant's need and desire to preserve trade secrets in computer software. Therefore, the Office allows authors to deposit incomplete printouts of software programs. An applicant can deposit the first and last twenty-five pages, for example, of a long program in order to preserve the trade secret status of the intervening pages.<sup>117</sup> Program designers find this rule useful because it is the middle of a lengthy program that is often most valuable. However, practitioners have difficulties identifying the first and last twenty-five pages of object-oriented programs because these programs are not linearly structured and consist of a collection of many sub-programs. Some guidance from the Copyright Office regarding object-oriented programs would benefit attorneys and clients alike.

The current pronouncement of the Copyright Office suggests that an applicant should only submit one registration for each computer program.<sup>118</sup> Apparently, authors were submitting several registrations that would collectively cover both the literal and

---

115. The Patent Office accepts, and indeed requests, a microfiche copy of programs that are longer than ten pages. See 37 C.F.R. § 1.96 (1976). See also MPEP, *supra* note 71, § 608.05.

116. 37 C.F.R. § 202.20 (c)(2)(viii)(A)(2) (1995). An applicant can deposit a printout of the object code of a program rather than the source code under the Office's "rule of doubt" form of registration. See 37 C.F.R. § 202.20(c)(2)(vii)(B) (1995). A registration will issue for the object code, but will not serve as *prima facie* evidence of copyrightability.

117. See 37 C.F.R. § 202.20(c)(2)(vii)(A)(1). Otherwise, more than half of the work needs to be deposited in one form or another. 37 C.F.R. § 202.20(c)(2)(vii)(A)(2).

118. See Copyright Registration for Computer Programs, United States Copyright Office Circular 61 (Nov. 1993).

non-literal aspects of a single program. The different registrations would include different sub-programs and subroutines, as well as the output of a computer program and the display screens that the program generates. The Copyright Office limited this practice to one registration per work of authorship.

The advantage of this policy is that it has reduced the number of registrations that the Office is receiving. However, the negative effect of this policy is that it coerces applicants to decide what is the most original and creative aspect of each software work. Each computer program consists of original textual elements, and often original audio-visual elements as well. The current pronouncement forces the author to determine whether the work is primarily textual or primarily audio-visual in nature.<sup>119</sup> The Copyright Office should consider allowing applicants to file two registrations for each program, one covering the literal text of the program itself and the other covering any non-literal output generated thereby. Such a policy would help to clarify the extent of protection accorded software under the copyright laws.

### III. TRANSACTIONAL ISSUES

The transactional issues involved in the area of computer software protection primarily effect a buyer who purchases a program from a retailer, and a person who hires another to develop or write the program itself. Over the years, these transactional issues have given rise to disputes which the legislature and the courts have failed to completely resolve. The discussion of this section centers around problems that the industry has faced with "shrink-wrap" licenses. This section concludes with a brief discussion of "work made for hire" agreements.

Shrink-wrap licenses are licenses that accompany a computer program that is sold at a retail store or "off the shelf." For example, a prepackaged program sold on one or more floppy disks is typically enclosed in an envelope bearing some fine print on either or both sides. This fine print is the shrink-wrap license. This license usually states that upon opening the envelope, the buyer accepts the license and agrees not to copy, transfer, license, sell, or develop derivations of, the software. Essentially, these licenses are unilateral contracts between the buyer and seller of commercial software. Because the seller writes the terms of the license, it typically includes very restrictive terms such as prohibitions on copying the software.

In the 1980s, several states adopted laws that were designed to enforce these shrink-wrap licenses. Illinois adopted a shrink-

---

119. *Id.* If, however, an applicant is in doubt he or she can include both the program text and its output as part of a single registration. *Id.*

wrap law in 1985.<sup>120</sup> In 1987, however, Louisiana's shrink-wrap law<sup>121</sup> ran afoul of the Copyright Act.<sup>122</sup> In *Vault v. Quaid*, the District Court for the Eastern District of Louisiana struck down the Louisiana statute because it violated the limited copying that the Copyright Act permitted.<sup>123</sup> The district court found that because the Copyright Act allows copying for archival purposes, and the Louisiana shrink-wrap law prohibited making any copies of the purchased software, the statute was invalid.<sup>124</sup> After *Vault v. Quaid*, many states, including Illinois, began repealing their shrink-wrap statutes.<sup>125</sup> However, the practice of using these unique licenses is still in force today, and questions remain as to whether these licenses are enforceable.

The District Court for the Western District of Wisconsin recently revisited this issue in *ProCD v. Zeidenberg*.<sup>126</sup> The *Zeidenberg* court refused to enforce the ProCD shrink-wrap license explaining that it was a unilateral contract.<sup>127</sup> According to the court, the purchaser of the software did not acquiesce or agree to the terms of such a license.<sup>128</sup> On appeal, however, the Seventh Circuit upheld the license as a valid contract.<sup>129</sup>

Despite the case law in this area, questions continually surface regarding the use and enforceability of these licenses. Practitioners have proposed modifications of the Uniform Commercial

---

120. ILL. REV. STAT. ch. 29, para. 801-808 (1985).

121. LA. REV. STAT. ANN. §§ 1962-65 (West 1984).

122. See *Vault v. Quaid*, 655 F. Supp. 750, 763 (E.D. La. 1987) (holding that the portions of the subject agreement that conflicted with the Federal Copyright Act were void because the Louisiana Software Act was preempted by an area of federal law).

123. *Id.* at 763. See 17 U.S.C. § 117 (1976) (allowing a copy or adaptation of a computer program if created as an essential step in the utilization of the program or for archival purposes).

124. *Id.*

125. The Illinois Software Enforcement Act, ILL. REV. STAT. ch 29, para. 801-808 (1985), was repealed by P.A. 85-254, § 1, effective Jan. 1, 1988; P.A. 85-614, § 1, effective Jan. 1, 1988.

126. *ProCD, Inc. v. Zeidenberg*, 908 F. Supp. 640 (W.D. Wis.), *rev'd and remanded*, 86 F.3d 1447, 1455 (7th Cir. 1996).

127. *Zeidenberg*, 908 F. Supp. at 655. The court explained that it would apply the provisions of the Uniform Commercial Code to transactions involving computer software. *Id.* at 650-51. Matthew Zeidenberg, a defendant in this case, had purchased a copy of a nation-wide directory of business and residential listings that ProCD Incorporated, the plaintiff, was selling on CD-ROM. *Id.* at 644-45. This product contained a shrink-wrap license which provided, in part, that "[b]y using the discs and the listings licensed to you[, the purchaser,] you agree to be bound by the terms of this License." *Id.* at 644. However, the shrink-wrap license was inside the product's box and Zeidenberg was not able to read the license prior to purchasing ProCD's product. *Id.* at 651. Therefore, the court held that Zeidenberg did not agree to the terms of the license. *Id.* at 655.

128. *Id.* at 655.

129. *ProCD, Inc. v. Zeidenberg*, 86 F.3d 1447, 1455 (7th Cir. 1996).

Code in order to address software licensing.<sup>130</sup> The proposed modifications may clarify these licensing issues and bring greater enforceability and legitimacy to shrink-wrap licenses. Nonetheless, it is now clear that attorneys and clients cannot ignore the limitations of the Copyright Act in drafting shrink-wrap licenses.

Another transactional issue regarding computer software is the "work made for hire" provisions of the Copyright Act. Practitioners have questioned whether a writing is required to support a work made for hire agreement. The law provides that a written agreement is required to convey ownership of a copyright.<sup>131</sup> Therefore, in the hiring of outside developers or contractors, a written document is essential to transfer the copyright in any computer software developed by the contractor.<sup>132</sup> For employees that write programs within the scope of their employment, however, it is the employer who will own the copyright to these works in the absence of any agreement to the contrary.<sup>133</sup>

#### IV. PROPOSALS FOR THE FUTURE

While the difficulties that currently exist in the area of computer software protection are pervasive, there are some solutions to these existing problems. For example, the creation of a separate federal software statute would be a step in the direction towards averting some of the problems that continue to challenge practitioners in the field. A federal software statute may bridge the gap that currently exists between patent and copyright protection for computer software. On one side of the gap is a concern over the time that it takes to obtain patent protection balanced against the broad scope of protection accorded by the patent laws. On the other side of the gap is a concern over the time and scope of protection under the copyright laws balanced against the low cost of protection. Choosing the most appropriate form of protection for a given situation involves a cost/benefit calculus of each form of protection.

There is, however, a middle ground that the legislature has not yet covered, and a separate software statute can help plug the gap. This middle ground would provide a shorter term of protec-

---

130. See U.C.C. Art. 2, Ch. 3 (Proposed Draft Sept. 10, 1994) (explaining that the proposed draft of Article 2 dealt with the treatment of software contracts).

131. See 17 U.S.C. § 204(a) (1976).

132. *Id.* See 17 U.S.C. § 202 (1976) (providing that ownership of a copyright is distinct from ownership of any material object embodying the work, and transfer of the object does not itself convey transfer of the copyright).

133. See, e.g., *Community for Creative Non-Violence v. Reid*, 490 U.S. 730 (1989); *Roeslin v. District of Columbia*, 921 F. Supp. 793 (D.D.C. 1995); *Miller v. CP Chems., Inc.*, 808 F. Supp. 1238 (D.S.C. 1992); *Avtec Sys., Inc. v. Peiffer*, 805 F. Supp. 1312 (E.D. Va. 1992), *rev'd and remanded*, 21 F.3d 568 (4th Cir. 1994).

tion than is currently afforded under the patent and copyright laws. The term should roughly coincide with the product life or shelf life of computer software — a term of protection in the three to five year range would be sufficient. This term of protection would solve some of the problems that currently exist in this continuously changing area. However, the statute should only cover the non-literal aspects of a computer program — an area to which traditional copyright laws may no longer extend protection. Further, the statute should cover and protect only the novel features of that program, thus stopping short of what the patent laws already address (i.e., novel and non-obviousness inventions).

One question that needs to be addressed is the process of acquiring or perfecting one's rights. Should the process more resemble patent prosecution or copyright registration? Prosecution for patents takes time and costs a great deal of money. Conversely, registration for copyrights is very cheap and quick. A limited application practice that more closely resembles copyright registration, with a short form of examination to confirm whether there are novel and original aspects to the program, would be suitable for this middle ground of protection. The provisions of the Patent Act would remain to provide protection for not only novel, but non-obvious programs. More than mere originality, however, would be needed to gain protection under the new statute, thus adding a novelty requirement over and above traditional copyright protection.

The scope of protection afforded software would need to be narrower than the protection of the Patent Act, but still preserve a prohibition against independent development. Thus, broader protection would be available over the narrow prohibition against copying provided under the Copyright Act. The details of a software statute can be addressed should a legislature consider the enactment of such a law. The industry needs this type of protection, however, in order to remain competitive both domestically and globally. The question lingers as to whether Congress will address this concern in any future legislation that it may propose.

#### CONCLUSION

For the future, it appears as though the copyright law is returning to literal protection only for computer software, reserving the broad protection of the non-literal elements of a computer program for the patent laws. The PTO and the CAFC are finally humming the same tune with respect to patentable subject matter. Hopefully, the Patent Office will overcome its internal problems quickly, and assemble and train an examining core that can properly handle these unique and complex applications in a minimum amount of time. The Office must soon tackle and relieve the huge backlog of software-related applications that it has accumu-



lated.

Within this framework, the Internet will continue to push the present protection schemes to the limit. Authors are currently developing new software to take advantage of this medium, and it appears that Congress will do little to intervene.<sup>134</sup> Therefore, it remains for the Patent Office, the Copyright Office, and the courts to hammer out a scheme of protection for computer software within the legal framework that currently exists. While 1995 did not advance the protection of computer software significantly, one can only hope that the future will better clarify the available protection for computer software — if not for the practitioner's sake, at least for the client's sake.

---

134. See Intellectual Property and the National Information Infrastructure, The Report of the Working Group on Intellectual Property Rights <<http://www@uspto.gov>> (last modified Nov. 15, 1995).