Loyola of Los Angeles
Entertainment Law Review

Volume 24 | Number 2 Article 1

3-1-2004

Let's Swap Copyright for Code: The Computer Software Disclosure Dichotomy

Christina M. Reger

LET'S SWAP COPYRIGHT FOR CODE: THE COMPUTER SOFTWARE DISCLOSURE DICHOTOMY

*Christina M. Reger**

I. INTRODUCTION

One man's jumble of 001 0000011110010110 1101 is another man's object code. Similarly, to the ordinary person unskilled in computer language, "depress file," "highlight toolbar," "drop down box," "move cursor," "move highlighted box," "depress print," "highlight print," "send message to queue printer," and "display printer options" sound like elementary printing instructions, while a computer programmer recognizes these as potential commands for printing. Congress classified these creations, object code and source code respectively, as literary works protectable under the 1976 Copyright Act ("the 1976 Act").[1] The 1980 amendments to the 1976 Act ("the 1980 amendments") incorporated computer programs, comprised of both object code and source code, and classified them as literary works.[2] Once something is classified as a literary work, it receives all of the benefits of copyright protection granted upon registration.[3]

Under the 1976 Act, copyright commences the moment a work is "fixed in any tangible medium of expression."[4] However, certain privileges, including the right to sue for infringement,[5] come only when a

---

* To my Father for teaching me that anything is possible, and to my Mother for her continued love, friendship, and support. Christina Reger earned her J.D. from Seton Hall University School of Law in 2003 and is currently practicing law with Budd Larner Rosenbaum Greenberg & Sade, P.C. in New Jersey.

1. *See* Copyright Act of 1976, 17 U.S.C. §§ 101–122 (2000).

2. *See* Pub. L. No. 96-517, § 10(a), 94 Stat. 3028 (1980) (codified as amended at 17 U.S.C. §§ 101–122 (2000)). Congress incorporated computer programs into this Act. *Id.*

3. *See id.*

4. *Id.* § 102.

5. *See id.* §§ 411–412 (establishing registration as a prerequisite to an infringement action); *see also id.* §§ 501–511 (detailing the remedies available, including injunctions, monetary

copyright owner registers the work with the United States Copyright Office and deposits a "complete" copy of the "best edition" with the Library of Congress.[6] The deposit requirement provides for disclosure of the information contained in the work.[7] In fact, even as early as the 1909 Copyright Act,[8] the *quid pro quo* of federal copyright protection has been disclosure.[9] Disclosure stimulates and encourages creativity by enabling other authors to create and develop new works.[10] Additionally, disclosure promotes economic efficiency by building upon current works to create better, more efficient creations and processes.[11] Society, in return, receives a creative work that stimulates and promotes future works of greater efficiency.

However, because the process for obtaining a copyright for computer programs significantly differs from that for all other literary works, the copyright law's application to computer programs is flawed.[12] Federal copyright protection for computer programs, both object and source code,

---

damages, seizure, and destruction of infringing material).

6. *See id.* §§ 408, 412. The Copyright Act defines "best edition" as "the edition, published in the United States at any time before the date of deposit, that the Library of Congress determines to be most suitable for its purposes." *Id.* § 101. One familiar with computer code, and the thousands of pages it produces, may surmise this definition is one of convenience. "Complete" is not defined in the Copyright Act. Webster's Dictionary defines "complete" as "having all necessary parts, elements, or steps." MERRIAM-WEBSTER'S COLLEGIATE DICTIONARY, 235 (10th ed. 1998). One might suggest these two definitions somewhat contradict each other. *See also* 37 C.F.R. § 202.20(c)(vii)(A)(1)–(2) (2002) (describing deposit requirements). In contrast, patent statutes require complete disclosure of all material relative to patentability to best serve the public interest and provide specific statutory language regarding disclosure of all relevant information. 37 U.S.C. §§ 1.52, 1.56, 1.58 (2002) ("The public is best served, and the . . . patent examination [is most effective] when, . . . the Office is aware of and evaluates the teachings of *all* information."). 37 U.S.C. § 1.56 (2002) (emphasis added). In fact, the statutes further require that the files will be open to the public. 37 U.S.C. §§ 1.11, 1.12 (2002). *See also* the Rules of Practice for filing Trademarks, which require a description of the mark, a drawing, and a specimen. 37 U.S.C. §§ 2.35, 2.51, 2.52, 2.56 (2002).

7. *See* 37 C.F.R. § 202.20 (describing the general requirement as a *complete* copy).

8. Copyright Act of 1909, 17 U.S.C. § 13 (1947) (requiring deposit of the best edition).

9. *See id.*

10. *See* Stephen M. McJohn, *The Paradoxes of Free Software*, 9 GEO. MASON L. REV. 25, 38 (2000); *see also infra* Part V.

11. McJohn, *supra* note 10, at 38–40 (discussing that all intellectual property law should promote economic efficiency).

12. *See generally* 17 U.S.C. §§ 101–122 (codifying the process for obtaining copyright protection for computer programs and literary works).

is an anomaly within the provisions for obtaining federal protection.[13] In essence, computer code personifies the "golden child" in the field of copyright law because computer code requires *limited* disclosure, yet receives the same privileges.[14]

Source code is written by highly technical individuals called programmers or developers.[15] This code is subsequently converted into object code, which makes up a computer program or software package.[16] Since the creation of source code is costly and time consuming, it is extremely valuable. The traditional model for the development of computer programs is referred to as a "closed source model," in which the source code is held in secret and the program is distributed in its nonsensical object code form, which only the program understands.[17]

An author is not required to submit a *complete* copy of the source code to register for copyright protection.[18] Rather, only a portion of the work, customarily the first and last twenty-five pages (with some variations), must be submitted.[19] Furthermore, the 1976 Act permits an author to block out work containing trade secret information within the submitted pages.[20] In exchange for this limited disclosure, the author receives the benefits of copyright registration; society receives nothing.

The open source model, an alternative form of software development, meets not only the typical disclosure requirements[21] but also fulfills the goals of copyright protection.[22] Commonly known as "copyleft," the open source model promotes access to copy, modify, and distribute versions of software, on the condition that all subsequent versions remain free for all users.[23] This model promotes efficiency by eliminating reverse

13. *See id.*

14. *See id.*

15. *See* Wilson at 30.

16. *See* Wilson at 30.

17. *See* McJohn, *supra* note 10, at 36.

18. 37 C.F.R. § 202.20(a)(2)(ii).

19. *See id.* § 202.20(A)(1).

20. *See id.* § 202.20(A)(2).

21. Typically meaning the submission of a complete copy of the best edition as required in 17 U.S.C. § 408.

22. *Id.*

23. Richard Stallman, *The GNU Operating System and the Free Software Movement, in* OPEN SOURCES: VOICES FROM THE OPEN SOURCE REVOLUTION, 59 (Chris DiBona et al. eds.,

engineering[24] and facilitates reuse, while simultaneously encouraging derivative works and stimulating new creations.[25]

Part II of this Article examines the fundamentals of computer software and its structure under the traditional closed source.[26] Part III briefly explores the laws that protect computer programs in general and analyzes the fundamental principles of copyright law as Congress has designated copyright law the primary mode of federal protection for computer programs. Part III also reviews the registration requirements for obtaining copyright protection for computer programs. Part IV provides an in-depth analysis of the dichotomy in the registration process between computer programs and other literary works, and discusses how this process contradicts the fundamental principles of copyright law. Part V of this Article proposes the open source model as an alternative to this dichotomy. Finally, part VI summarizes the inadequacies of the current protection for computer programs and advocates open source as the viable solution.

II. WHAT IS COMPUTER SOFTWARE?

Computer software is a set of instructions or data that a computer follows, acts upon, lists, or displays for the user's benefit.[27] This includes computer programs, documentation, databases, and user interfaces.[28] Software subdivides into three components: literal code, non-literal code, and structural components, all of which enable the program to produce its desired results.[29] Literal code, deemed protectable under the 1976 Act, is the written text and computer instructions.[30] Non-literal code is the

---

1999).

24. Page 106 right before section III.

25. *See id.* at 158–69.

26. The industry also utilizes patent and trade secret laws to protect software.

27. David W. Carstens, *Legal Protection of Computer Software: Patents, Copyrights, and Trade Secrets*, 20 J. CONTEMP. L. 13, 15 (1994).

28. *Id.*

29. *See* Julian Velasco, *The Copyrightability of Nonliteral Elements of Computer Programs*, 94 COLUM. L. REV. 242, 242 (1994).

30. *See* 17 U.S.C. § 101 (2000) (defining a "computer program" as "a set of statements or instructions to be used directly or indirectly in a computer . . . to bring about a certain result"); *see also id.* § 106 (detailing the exclusive rights of copyright owners).

translation or sequencing of such text to achieve a desired result.[31] Its protection is the source of much litigation and commentary.[32]

Computer code exists in three formats: flowcharts, source code, and object code.[33] Programmers initially draft a new program in a flowchart format.[34] The flowchart embodies the idea of the program.[35] Using the flowchart, the programmer then writes the source code[36] in a high-level programming language, such as BASIC, FORTRAN, or PASCAL, which corresponds with the spoken English language.[37] Source code includes primarily descriptive words, formulas, and mathematical equations.[38] Once the source code is complete, a compiler translates the written source code into "executable" code, i.e. object code.[39] Object code is a low-level computer language that is generally unintelligible.[40] Object code consists primarily of binary ones and zeros read by the computer to run the program.[41]

To date, writing computer code has largely been a manual process.[42]

---

31. *See* Velasco, *supra* note 29, at 242–48; *See also* Himanshu S. Amin, *The Lack of Protection Afforded Software Under the Current Intellectual Property Laws*, 43 CLEV. ST. L. REV. 21–22 (1995).

32. The federal courts proffer four distinct tests to determine copying of non-literal elements. *See* Whelan Assocs. v. Jaslow Dental Labs., 797 F.2d 1222 (3d Cir. 1986) (broadening protection to classify the idea as the purpose of the program while the remaining program obtains protection as expression of the idea); *see also* Lotus Dev. Corp. v. Paperback Software Int'l., 740 F. Supp. 37 (D. Mass. 1990) (explaining three-part test requiring a decision maker to separate the idea from the expression and to determine the copyrightability of the essential elements); *see also* Brown Bag Software v. Symantec Corp. 960 F.2d 1465 (9th Cir. 1992) (developing the Extrinsic-Intrinsic test for substantial similarity); *see also* Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 711, 717 (2d Cir. 1992) (developing the abstraction/filtration/comparison test). *See generally* Velasco, *supra* note 28, at 242–44 (discussing copyright issues and non-literal elements).

33. *See* Carstens, *supra* note 27, at 15–16.

34. *See id.* at 16.

35. *See id.*

36. *See* McJohn, *supra* note 10, at 26.

37. *See id.*

38. *See* Amin, *supra* note 31, at 21; *see also* Velasco, *supra* note 29, at 244.

39. Clyde H. Wilson, Jr., *Software Piracy Litigation*, 67 FLA. B.J. 29, 30 (1993); *see also* McJohn, *supra* note 10, at 26.

40. *See* Amin, *supra* note 31, at 21.

41. *See id.*

42. *See* Patrick K. Bobko, *Open-Source Software and the Demise of Copyright*, 27 RUTGERS COMPUTER & TECH. L. J. 51, 57–58 (2000).

Hence, the primary cost associated with the creation of software is labor.[43] Producing the first copy is extremely costly while reproduction costs are virtually miniscule.[44] Because of the exorbitant creation costs, computer programmers and software distributors closely guard the source code and distribute only the object code.[45] Accordingly, when consumers or manufacturers purchase software, they receive the program in object code.[46]

Distributing the program exclusively in object code reduces the risk of exposing the source code, because[47] disclosing the source code would enable another person to recreate and potentially sell and profit from the program.[48] Nevertheless, a consumer or competing manufacturer may attempt to translate the code back into source code manually or with the assistance of a decompiler.[49] This process, known as reverse-engineering, produces the source code, but is extremely costly, time consuming, and error-laden.[50]

The process described above is often referred to as the closed source model, which most companies operate under today.[51] Economically, this model operates on two assumptions: (1) selling the product will compensate the company for the developer's time and labor, and (2) the market price of the software will be proportionate to its economic value.[52] Therefore, by selling the software, the company will theoretically recover

---

43. *See id.* at 56–58 (arguing that the majority of the costs associated with software development are incurred up front). For example, Microsoft incurred costs of nearly one million dollars to produce Windows 2001. *Id.* at 61. The development required the use of over 5000 processors over a four-year period. *Id.*

44. *See id.* at 59.

45. *See* Ronald L. Johnston & Allen R. Grogan, *Trade Secret Protection for Mass Distributed Software*, THE COMPUTER LAW. Nov. 1994, at 1.

46. *See id.*

47. *See id.* at 1–2.

48. *See* Carstens, *supra* note 27, at 17 (noting that others can copy software with minimal expenditure).

49. *See* McJohn, *supra* note 10, at 36.

50. *See id.* In addition to the time consuming nature of the reverse engineering process, McJohn asserts that it is an uncertain enterprise. *Id.* Software giants such as Microsoft protect themselves by continuously upgrading their product, making reverse engineering software perpetually out of date. *Id.*

51. *See id.* Microsoft represents the successful application of this theory.

52. Bobko, *supra* note 42, at 60.

its production costs.

III. LEGAL PROTECTION AVAILABLE FOR COMPUTER SOFTWARE

Currently, there are three prominent methods available for protecting the various components to computer software: copyright, patent, and trade secret.[53] Copyright law protects the literary work, the computer code;[54] patent law safeguards the utilitarian aspects;[55] and trade secret law theoretically protects the idea.[56] As a written work with a utilitarian purpose,[57] computer programs deride and defy categorization in the present library of intellectual property protection.[58] Despite this paradox, Congress designated copyright law as the method of protection for computer program, or code, bringing it under the umbrella of federal protections.[59] Consequently, works are protected for what they express (the tangible medium) and not how the expression results in operation (the utilitarian purpose).[60]

A. The Evolution of Protection for Computer Programs Under the 1976 Copyright Act

Copyright protection originates from the United States Constitution.[61] The Constitution grants Congress the power "[t]o promote the Progress of Science and the Useful Arts by securing for limited Times to Authors and Inventors the exclusive right to their respective Writings and

53. *See* Carstens, *supra* note 27, at 17 (summarizing these types of protections).

54. *See* 17 U.S.C. §§ 102, 106 (2000).

55. Generally, patent law is an exclusionary right that provides protection for any new and useful process and is codified in 35 U.S.C. §§ 101, 154 (2000); *see also* Baker v. Selden, 101 U.S. 99, 104 (1879). The protection is short-term (currently 20 years) and is conditioned on the object being novel, useful, and non-obvious. *See id.* §§ 103, 154.

56. Trade secret law is primarily state law, generally derived from the Uniform Trade Secret Act, and offers protection for ideas, processes, logic, and engineering of computer programs. UNIF. TRADE SECRET ACT §§ 1–11 (amended 1985), 14 U.L.A. 437 (2001); *see also* Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 711, 717 (2d Cir. 1992) (noting that trade secret protection is explicitly available for the gaps in protection under copyright law).

57. Processes with utilitarian purpose are protectable under patent law. 35 U.S.C. §§ 101, 154 (2000). Patent law protects a computer program's functional aspects—the inventive idea. *See also See also* Amin, *supra* note 31, at 19, 21.

58. *See Baker*, 101 U.S. at 105.

59. *See generally* 17 U.S.C. §§ 101–106 (1980 Amendment added definition of "computer program" to the regime).

60. *See Baker*, 101 U.S. at 105.

61. U.S. CONST. art. I, § 8, cl. 8.

Discoveries."[62] Copyright law embodies this Constitutional intention.

Copyright law advances two competing policy principles: to provide creators and authors with an economic incentive to produce, while limiting the protection that is granted to preserve the public domain in a manner such that the granted monopoly does not prevent others from conceiving and developing similar works.[63] As the court in *Computer Associates International, Inc. v. Altai, Inc.* stated, "[c]reative work is to be encouraged and rewarded, but private motivation must ultimately serve the cause of promoting broad public availability of literature, music, and the other arts."[64] The economic theory behind copyright law is the advancement of public welfare through motivation of individual creation for personal gain.[65] Thus, courts must balance granting a limited monopoly that provides a return on individual investment against permitting other authors the use of basic building blocks.[66]

The 1976 Copyright Act replaced the 1909 Copyright Act[67] and defines copyrightable subject matter as "original works of authorship, fixed in a 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."[68] This broad protection does not 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."[69] This principle emanates from the seminal case of *Baker v. Selden*, where the Supreme Court applied the existing copyright law to determine that a book depicting an accounting ledger was not copyrightable.[70] In so holding, the Court noted that accounting forms are outside the realm of copyright

62. *Id.*

63. Fred Anthony Rowley, Jr., Note, *Dynamic Copyright Law: Its Problems and a Possible Solution*, 11 HARV. J.L. & TECH. 481, 481 (1998).

64. 982 F.2d 693, 711 (2d Cir. 1992).

65. Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045, 1058 (1989).

66. *See id.*

67. *See* 17 U.S.C. §§ 101–106.

68. *Id.* § 102(a).

69. *Id.* § 102(b).

70. 101 U.S. 99, 107 (1879) (denying copyright protection for a book outlining an accounting system as lacking expression of an idea).

protection because the "illustrations and diagrams employed happen to correspond more closely than usual with the actual work performed by the operator who uses the art."[71] Although the Court found the essays and explanations of accounting principles protectable under copyright law, the Court classified the diagrams as utilitarian and possibly protectable under patent law.[72] The gap in protection in protection between an idea and the expression of that idea into a tangible medium later became known as the idea/expression dichotomy.

Under the 1976 Act, copyright extends from the moment the creative work becomes fixed in a tangible form.[73] However, to be able to sue for infringement[74] the author must register the work with the Copyright Office.[75] Also, an author cannot obtain statutory damages or attorneys' fees without a valid registration.[76]

Neither the 1909 Act, nor the 1976 Act, originally articulated computer programs as protectable subject matter.[77] In fact, the 1976 Act did not even include computer programs in the list of protectable works.[78] However, prior to passing the 1976 Act, Congress chartered the National Commission on New Technological Uses of Copyright Works ("CONTU") to examine new technologies and determine how to most appropriately

71. *Id.* at 104.

72. *See id.* at 104–05 (declining a discussion of whether the "art might or might not have been patented").

73. *See* 17 U.S.C. § 102(a) (providing copyright protection for works fixed in a tangible medium). "A work is fixed in a tangible medium of expression when its embodiment in a copy or phonorecord, by or under the authority of the author, is sufficiently permanent or stable to permit it to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration." *Id.* § 101.

74. *See id.* § 411(a). "[N]o action for infringement of the copyright . . . shall be instituted until registration of the copyright claim has been made in accordance with this title." *Id.*; *see also id.* § 501(b) (entitling the "legal or beneficial owner of an exclusive right . . . to institute an action for infringement . . .")

75. *See* Datastorm Techs., Inc. v. Excalibur Comm., Inc., 888 F. Supp. 112, 114 (N.D. Cal. 1995) (holding that registration is not a prerequisite for copyright protection but it is a prerequisite to file suit for infringement); *see also* Tang v. Hwang, 799 F. Supp. 499, 503 (E.D. Pa. 1992). In addition to exclusive ownership, to bring an action under the Copyright Act, the owner must register the work in accordance with the Act. *Id.*

76. 17 U.S.C. § 412.

77. *See* Copyright Act of 1909, 17 U.S.C. § 5 (1947), *amended by* 17 U.S.C. § 102.

78. *See* 17 U.S.C. § 102. Section 102 specifies categories of works of authorship, including protection for literary works, motion pictures, sound recordings, and most recently architectural works. *Id.*

revise the current intellectual property laws to provide protection for these technologies.[79] CONTU's mission was to incorporate computer programs and information technology into the existing menu of intellectual property protection.[80] After extensive research, CONTU recommended that the principal protection for computer software should reside within copyright law, notwithstanding that copyright law did not protect an "idea, procedure, process, system, method of operation, concept, principle or discovery."[81] Congress responded by implementing the recommendations offered by CONTU into the 1980 amendments to the 1976 Act.[82]

The 1980 amendments to the 1976 Act define a computer program as a set of statements utilized directly or indirectly by a computer to produce a specified result.[83] Pursuant to the 1980 amendments, authors of computer programs receive substantially the same protection as authors of other literary works.[84] These rights include the right to reproduce, distribute, and create derivatives of the work.[85] Again, as with other copyrighted works, Congress limited this protection to the *expression* of the idea, and not to the idea itself.[86]

Applying the 1976 Act, as amended, to computer programs has proven challenging because of the nature of the work.[87] Courts continue to experience difficulty in classifying what is "idea" and what is "expression" of the idea for purposes of determining copyright infringement.[88] As Learned Hand noted, "[n]obody has ever been able to fix that boundary,

79. *See generally* NAT'L COMM'N ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REP. (1979) at 3 (explaining the reason for the commission's creation and the results of its inquiry).

80. *See id.*

81. *See generally id.* Commissioner Nimmer cautioned that CONTU's recommendations might force copyright law to "the breaking point," transforming it to a general misappropriations law. *Id.*; *see also* 17 U.S.C. § 102(b) (1982).

82. *See* Act of Dec. 12, 1980, Pub. L. No. 96-517, § 10(b), 94 Stat. 3007, 3028 (codified as amended at 17 U.S.C. §§ 101, 117 (2001)).

83. 17 U.S.C. § 101.

84. *Id.* §§ 101–102, 106.

85. *Id.* § 102(a).

86. *Id.* § 102(b).

87. *See generally* Carstens, *supra* note 27 (detailing various approaches and problems in applying copyright to computer programs).

88. *See* Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied*, 282 U.S. 902 (1931).

[between unprotectable idea and protectable expression,] and nobody ever can."[89]

The Third Circuit was one of the first courts to grapple with the distinction of whether the underlying structure of a computer program was copyrightable. In *Whelan Associates Inc. v. Jaslow Dental Laboratory Inc.*, the Third Circuit took an expansive view in identifying copyrightable content for computer programs.[90] The court held "[t]he purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of that idea."[91] Thus, any structure, sequence, or organization not essential to the purpose was considered expression and therefore copyrightable .[92]

Responding to the broad protections established in *Whelan*, the Second Circuit crafted a test to deflate some of the protection established by the *Whelan* court. In *Computer Associates International, Inc. v. Altai, Inc.*, the Second Circuit developed the abstraction/filtration/comparison test.[93] This three-part test determined the appropriate level of abstraction to apply, filtered any non-copyrightable elements, and compared the remaining elements for alleged infringement.[94] Overall, the Court in *Altai* affirmed that the current copyright laws intend to protect computer programs but "only to the extent that they incorporate authorship in [the] programmer's expression of original ideas, as distinguished from the ideas themselves."[95]

Later courts continued to expand this protection, holding that computer programs are protectable under the 1976 Copyright Act in both source and object code.[96] However, notwithstanding the recognition of computer programs as copyrightable, courts continue to grapple with the idea/expression dichotomy as a precursor to their copyright infringement analysis. For example, in *Apple Computer, Inc. v. Franklin Computer*

89. *Id.*

90. 797 F.2d 1222, 1237–39 (3d Cir. 1986).

91. *Id.* at 1236.

92. *See id.*

93. 982 F.2d 693, 706–11 (2d Cir. 1992).

94. *See id.*

95. *Id.* at 703.

96. *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1255 (3d Cir. 1983).

Corp., the court held that copyrightable computer programs written in object code and embedded in read only machine (ROM) language are copyrightable.[97] The court analyzed the idea/expression dichotomy to ensure "the preservation of the balance between competition and protection [was] reflected in the . . . copyright laws."[98] In concluding that object code is copyrightable, the court held that since an idea can be expressed in various ways, the program embedded in ROM is an expression of an idea and hence copyrightable.[99]

Later courts willingly accepted that source and object codes are copyrightable but cautioned against too broad an application of this premise. In *Quinn v. City of Detroit*, an employee of the city developed a custom software program from an "off-the-shelf" software package.[100] The city petitioned for summary judgment, arguing that the employee did not possess a valid copyright in his program because he did not "write" it, but merely customized a pre-existing copyright.[101] In holding that a computer program's source and object code are both copyrightable, the court denied the city's motion, noting that an issue of fact existed as to whether the employee possessed a valid copyright.[102] Additionally, the court cautioned against potential future implications of the copyrightability of custom application as programs become easier to use and manipulate.[103]

Once a court successfully navigates through the idea/expression classifications, it must determine if infringement has occurred. Here again, copyright law provides an imperfect fit. In *Sega Enterprises Ltd. v. Accolade, Inc.*, the court applied a rather untraditional approach of the fair use analysis and held that Accolade's disassembly of Sega's computer code to create a competing product was not copyright infringement.[104] In so holding, the court concluded that the disassembly of the code gave

97. *Id.* at 1249.

98. *Id.* at 1253 (quoting Herbert Rosenthal Jewelry Corp. v. Kalpakian, 446 F.2d 738, 742 (9th Cir. 1971)).

99. *See* id. *Cf.* Freedman v. Grolier Enters., Inc., 179 U.S.P.Q. BNA 476, 478 (S.D.N.Y. 1973) ("Copyright protection will not be given to a form of expression necessarily dictated by the underlying subject matter.").

100. 988 F. Supp. 1044, 1046 (E.D. Mich. 1997).

101. *Id.* at 1054.

102. *Id.*

103. *See id.*

104. 977 F.2d 1510, 1527–28 (9th Cir. 1993).

Accolade access to the underlying ideas and functional elements, which are not protectable under copyright.[105] The resulting doctrine again moved the proverbial line of copyrightability to favor unprotected ideas rather than protectable expression in computer code re-engineering.[106] Later courts have declined to further extend this analysis.[107]

Although courts today seem to blindly accept all source and object code as being copyrightable,[108] the interpretation of the 1980 amendments in determining the fine line of what constitutes idea versus expression for purposes of infringement has produced uncertain and inconsistent results.[109] Uncertainty impedes the risk-taking inherent in innovation, and this impedes the purpose of the Copyright Clause of the Constitution-"to promote the Progress of Science and the useful Arts."[110] Additionally, new issues in providing adequate protection continue to challenge the courts, as technology continues to outpace laws intended to protect that technology, and programs become easier to apply and manipulate.[111]

In most cases, courts have been unable to provide general, consistent rules to determine protection for computer programs. Recent decisions manifest the inability of copyright law to embody technological advances in software.[112] This quagmire of uncertainty and instability is a risk that software companies must consider when deciding whether to invest in or create new products.[113]

---

105. *See id.* at 1527.

106. *See* Mark A. Lemley & David W. O'Brien, *Encouraging Software Reuse*, 49 STAN. L. REV. 255, 279 (1997).

107. DSC Communications Corp. v. Pulse Communications, Inc., 170 F.3d 1354, 1363 (Fed. Cir. 1999).

108. *See, e.g.* Data Gen. Corp. v. Grumman Sys. Support Corp., 834 F. Supp. 477, 484 (D. Mass. 1992) (holding object code protectable under copyright laws); Fonar Corp. v. Domenick, 105 F.3d 99, 104 (2d Cir. 1997) (holding that "the literal elements of computer programs, i.e., their source and object codes, are the subject of copyright protection") (quoting *Computer Assocs. Int'l*, 982 F.2d at 702); *see* Bernstein v. United States Dep't of State, 922 F. Supp. 1426, 1436 (N.D. Cal. 1996) (holding that copyright protection extends to both source and object code).

109. *See* Lemley, *supra* note 106, at 258.

110. *See* U.S. CONST. art. I, § 8, cl. 8.

111. Marci A. Hamilton & Ted Sabety, *Computer Science Concepts in Copyright Cases: The Path to a Coherent Law*, 10 HARV. J.L. & TECH. 239, 245 (1997).

112. Michael Risch, *How Can Whelan v. Jaslow and Lotus v. Borland Both Be Right? Reexamining the Economics of Computer Software Reuse*, 17 J. MARSHALL J. COMPUTER & INFO. L. 511, 514–16 (1999).

113. *Id.* at 519.

B. Obtaining Copyright Protection—The Registration Process

Generally, an author must register their work with the United States Copyright Office to obtain federal copyright protection.[114] This requires the author to submit a completed application accompanied by a filing fee and a deposit of the work.[115] The deposit requirements are set forth in Title 37 of the Code of Federal Regulations.[116]. Although the deposit requirements vary, they all generally require a *complete* copy of the best edition of the published or unpublished work.[117] Therefore, to obtain federal copyright protection, the author must submit "all elements comprising the unit of publication of the best edition of the work, including elements that, if considered separately, would not be copyrightable subject matter."[118]

The deposit requirements for computer software, however, substantially deviate from these general requirements,[119] yet computer software is granted the same protection as that of other copyrightable works under the general rules. The 1976 Act, as amended, grants protection to unpublished computer programs (generally, source code) that are fixed, or, for programs that are published in machine-readable code only (object code).[120]

To obtain that protection, the author must deposit a copy of "identifying portions" of the work that are "visually perceptible without the aid of a machine or device."[121] The regulations define "identifying

114. *See* 17 U.S.C. § 411(a) (2000) "No action for infringement of the copyright in any United States work shall be instituted until registration of the copyright claim has been made in accordance with this title." *Id.*

115. *See* 37 C.F.R. § 202.3 (2002); *see also* 17 U.S.C. § 409.

116. *See generally* 37 C.F.R. §§ 202.19–202.20 (stating the deposit requirements for such items as motion pictures, lectures, holograms, and computer programs).

117. 17 U.S.C. § 408; *see also* 37 C.F.R. § 202.3(b), *see also* 37 C.F.R. § 202.20. For example, a deposit, for unpublished works, represents the *complete content* of the work. *Id.* § 37 C.F.R. § 202.20(b)(2)(i).

118. 37 C.F.R. § 202.19(b)(2).

119. *See id.* § 202.20(c)(2)(vii).

120. *Id.*

121. *Id.* § 202.20(c)(2)(vii)(A); *see also* Marybeth Peters, *Developments in the Copyright Office—Registration Practices, Address Before the Computer Law Association, Inc.* (Oct. 15–16, 1981) *in* COMPUTER SOFTWARE PROTECTION: A PRAGMATIC APPROACH, 1982, at 128–29. In 1964, when the Copyright Office first accepted registration for computer programs, claimants submitted the source code for deposit in accordance with the statute. *Id.* Then sometime later, the industry began submitting machine-readable object code. *Id.*

portions" as the first and last twenty-five pages of the source code.[122] For code containing trade secret information, the regulations provide several options. First, the deposit can consist of the first and last twenty-five pages of source code with the portions of the code containing trade secret information blocked-out.[123] However, the blocked-out portion must not proportionately exceed the remaining material.[124] Alternatively, the author can submit only ten pages of source code with no blocked-out portions, or twenty-five pages of object code together with ten pages of source code with no blocked-out portions.[125]

Thus, while J.K. Rowlings must submit all 870 pages of her latest best seller *Harry Potter and the Order of the Phoenix*[126] to obtain federal copyright protection, a computer programmer is only required to submit a limited number of pages to obtain that same protection.[127] Therefore, by only requiring the submission of this limited portion, federal copyright law grants a computer programmer all of the benefits and privileges of federal copyright protection.[128]

IV. THE DICHOTOMY OF COPYRIGHT PROTECTION FOR COMPUTER PROGRAMS

A. The Premise of Standard Discolsure

The cornerstone of the disclosure requirement originates in the Constitution, which established copyright protection to promote "Progress of Science and useful arts."[129] While it may be possible to achieve progress when information is kept secret, wide disclosure enriches society.[130] To

122. 37 C.F.R. § 202.20(c)(2)(vii)(A)(1) (describing programs without trade secrets within the source code).

123. *Id.* § 202.20(c)(2)(vii)(A)(2).

124. *Id.*

125. *Id.*

126. J.K. ROWLINGS, HARRY POTTER AND THE ORDER OF THE PHOENIX (Scholastic Press 2003).

127. *See* 37 C.F.R. § 202.20(c)(2)(vii).

128. *See generally id.* (discussing the "deposit" required to accompany an application for registration of a claim to copyright).

129. U.S. CONST. art. I, § 8, cl. 8.

130. *See* Dorothy M. Schrader, *Developments in the Copyright Office—Problems and Issues, Address Before the Computer Law Association, Inc.* (Oct. 15-16, 1981), *in* COMPUTER

encourage disclosure, the Constitution establishes a protective grant "for [a] limited [t]ime."[131] Thus, federal protection grants the owner the right to sue for infringement,[132] seek injunctive relief,[133] and receive monetary damages[134] in exchange for the disclosure that advances science and the arts. Thus, the very purpose of the disclosure requirement is fundamental to copyright law.[135]

Both the 1909 Act and the 1976 Act support and reinforce complete disclosure for the benefit of society as a whole.[136] For example, unpublished works were granted only common law protection under the 1909 Act.[137] Therefore, private papers and diaries could claim a common law copyright that would endure in perpetuity, yet authors were not required to disclose the documents to enrich society.[138] That regime actually hampered federal registration because an author claiming federal copyright did not have a superior right over an author claiming common law copyright.[139] This system encouraged concealment and discouraged disclosure of information.[140] The 1976 Act preempted state and common law copyright claims.[141] This change from the 1909 Act, granted access to the private papers and diaries previously protected under the common law regime, thereby eliminating common law copyright and creating access to

---

SOFTWARE PROTECTION: A PRAGMATIC APPROACH, 1982, at 126–27.

131. U.S. CONST. art. I, § 8, cl. 8.

132. 17 U.S.C. § 411 (2000).

133. *See id.* U.S.C. § 502; *see also id.* § 503 (granting court authority to order impounding or destruction of infringing articles).

134. *See id.* § 504 (allowing recovery of actual damages and infringer's profits); *see also* 17 U.S.C. § 505 (2001) (providing for reimbursement of reasonable attorneys' fees).

135. *See* U.S. CONST. art. I, § 8, cl. 8.

136. *See, e.g.*, Copyright Act of 1909, 17 U.S.C. § 13 (1947); *see also* 17 U.S.C. § 408 (2001).

137. *See* 17 U.S.C. § 2 (1947).

138. *See* H.R. REP. NO. 94-1476, at 130 (1976); *see also* STAFF OF HOUSE COMM. ON THE JUDICIARY, 89TH CONG., SUPPLEMENTARY REPORT OF THE REGISTER OF COPYRIGHTS ON THE GENERAL REVISIONS OF THE U.S. COPYRIGHT LAW: 1965 REVISION BILL, COPYRIGHT LAW REVISION PART 6 at 81–86 (House Comm. Print: 1965). *See* Schrader, *supra* note 130, at 126–27 (stating that the revisions provided scholars and researchers access to private papers and diaries once the information was released to the public). "One of the reasons for abolishing common law copyright was to force that kind of access." *Id.* at 127.

139. Schrader, *supra* note 130, at 126–27.

140. *Id.*

141. 17 U.S.C. § 301.

these papers after a number of years, once they were no longer kept secret.[142]

B. The Dichotomy In Protection For Computer Programs

The standard deposit requirement under the 1976 Act, which require complete disclosure, presented an immense problem for computer programmers seeking federal registration because they were required to expose their traditionally shrouded source code in exchange for the benefits of federal copyright.[143] In 1986, in response to this problem, Congress carved out an exception in the deposit requirements for computer programs, resulting in requirements that differ drastically from the requirements for all other literary works.[144] These particular requirements specifically account for the preservation of trade secrets by allowing portions of the code to be blocked out or withheld to avoid disclosure in the registration process.[145] In essence, these requirements allow programmers to submit a partial deposit while simultaneously securing secrecy for the code.[146] Despite this partial submission, the programmer receives the same federal copyright registration and protection given to all other literary works that require complete disclosure.[147] These special deposit requirements are antithetical to the very foundation of federal protection. Federal protection entails disclosure; this disclosure is the exchange for obtaining the federal right.[148]

There are various explanations for the divergence in the requirements

---

142. *See id.*

143. *See generally* 37 C.F.R. § 202.20(c)(2)(vii).

144. *See generally id.* (detailing the deposit requirements for all types of works).

145. *Id.* § 202.20(c)(2)(vii)(A)(2); *see also* Dominic Bencivenga, *Beyond Copyright Law: How to Protect Software,* NAT'L L. J., Apr. 22, 1996, at B1–B2 (referencing the availability of trade secret law for the "blocked out" portions of code).

146. *See generally id.* § 202.20(c)(2)(vii).

147. *See* 17 U.S.C. § 408 (establishing federal registration by paying a fee and depositing a copy of the work with the library of Congress). This section further sets forth the authority of the Register of Copyrights to regulate the nature of the deposited copies. *Id.*

148. *See generally id.* § 408(b)(2). All forms of federal protection require submission and often a detailed description of what is being protected. For example, for trademark protection, a submission requires a description of the trademark and/or a drawing, in addition to a sample generally affixed to the product. 15 U.S.C. § 1051(a)(2) (2001); *see also* 37 C.F.R. §§ 2.32, 2.51 (2001).

for computer programs. Lobbying efforts coupled with financial investments in the political process provide some explanations for the current structure of protection.[149] Individual computer programmers, commonly referred to as open source supporters,[150] have not actively lobbied Congress to re-examine the present incentives and inefficiencies created by the current protection regime.[151] By contrast, industry leaders like Microsoft actively lobby Congress to ensure the viability of the current structure.[152] In fact, Microsoft contributed over four million dollars to various election campaigns, making the company the fifth highest campaign contributor in the 1999–2000 election.[153] From this, one might conclude that money and persuasion influenced the current protection regime.

In addition to political contributions, industry leaders such as Microsoft, incorporate the current system into their corporate strategic goals.[154] Microsoft's strategy of "Embrace and Extend" enables it to retain control by "taking the result of open projects and standards, and adding incompatible Microsoft-only features in closed-source."[155] The resulting incompatible features and components require individuals and companies to continuously purchase upgrades for their products in order to maintain the same level of performance.[156] Recently, Microsoft attacked Linux, a program that is "open" or freely available for all users to see and manipulate, by sending Microsoft "spies" to its customers to determine if

149. *See* Computer Software: Top Contributors, *at* http://www.opensecrets.org/industries/contrib/asp?Ind=csizo&cycle=2000 (last visited Aug. 21, 2003).

150. *See infra* Part V.

151. *See* John Miano, *Programmers Are Programmed Against Unions*, Computerworld, *at* http://www.computerworld.com/printthis/2000/0,4814,54154,00.html (Nov. 20, 2000) (detailing the largely unsuccessful attempts to unionize programmers). "Programmers tend to be libertarian in their views and treat the two major political parties with suspicion, if not contempt." *Id.*

152. *See generally* Computer Software: Top Contributors, *supra* note 149. The report cited the Federal Election Commission figures where Microsoft donated more than $4.5 million to Democrats and Republicans in the 1999–2000 election period. This donation more than quadrupled the donation of the next highest donor. *Id.*

153. Computer Software: Top Contributors, *supra* note 149.

154. Bruce Perens, *Free Software Leaders Stand Together*, *at* http://freedevelopers.net/press/perens/ (last visited Aug. 1, 2003).

155. *Id.*

156. *Open Source Group Slams Microsoft*, WORLD REPORTER, May 16, 2001, *at* 2001 WL 7308692.

they were running any free software products.[157]

Although software owners can employ other means of protecting their creation, these methods are often more costly and require disclosure. For example, software owners could obtain a patent for the functional aspects of the program. However, many software owners avoid obtaining a patent, despite the considerable protection a patent affords, because patent law mandates a detailed disclosure of the process in exchange for the federal grant[158]—a tradeoff that most owners are often unwilling to make even though only some portions of the program are protectable under other laws.[159]

C. The Inefficiency Created By the Current System – A Constant Model of Reinvention

The current copyright provisions encourage one owner to one copyright. The copyright interest vests in the first person to reduce an idea to writing, file the registration, and make the deposit.[160] Accordingly, current copyright deposit provisions sacrifice efficiency and experience by demanding reinvention rather than reuse. In other words, a programmer will write, expand, modify and finally develop an efficient code that enables a basic program to function in a certain way. Another company can not use this basic code and build upon it, but rather must *reinvent* another way to accomplish the same function. In light of these provisions, some successful companies can support a business model that expends outrageous sums of money to be first and then spends additional funds only to make sure no one else copies, invents, or creates something substantially

157. *See generally* Mitch Wagner, *Gates Goes to War With Linux*, ITNEWS, (Sept. 9, 2003), *available at* http://www.itnews.com.au/story.cfm?ID=8799 (last visited Sept. 8, 2003). Brian Valentine, Senior Vice President of Microsoft Windows division, in a December 26, 2001 internal email communication, sent to inquire if Microsoft clients were operating on any Linux systems, stated, "We have to . . . dig deeper into your accounts!" *Id.* He further instructed the sales staff to ask probing questions, walk through client data centers, and find out client strategic plans and key projects. *Id.*

158. *See* 35 U.S.C. § 112 (2000) To obtain patent protection, the patentee must provide a detailed description of the process so that someone skilled in the art can replicate it. *See also* Amin, *supra* note 31, at 23.

159. *See* 35 U.S.C. § 112 (articulating the best mode requirement, which instructs the patentee to set forth the best method contemplated at the time the application is filed).

160. *See* 17 U.S.C. § 410.

similar.

However, two realities of computer code development contradict the justification for such a business model. First, programmers are people, not machines. People change jobs. Today's computer programmers are "young, mobile, and highly specialized."[161] Referred to as "'migrant workers of the high tech age," they move from competitor to competitor within the same industry.[162] In today's volatile job market, companies will pay top dollar for an experienced programmer, especially one employed by a competitor.[163] When these programmers move to another employer, the intellectual property knowledge they possess moves with them. However, once employed by a new company, the programmer must reinvent otherwise copyrighted code to avoid copyright infringement.[164] Realistically, although the programmer may be able to avoid literal copying of previously written code by recoding, the logical thought processes and non-literal structure originates from the same source and may manifest itself in later creations.[165] Thus, the resulting program may nonetheless be the same. As a result, the employer invests significant amounts of time, energy, and money in a programmer whose subsequent work may be the subject of a later copyright infringement action.

Secondly, because reinvention results in variation and change, and because people generally do not like change, continued reinvention may discourage consumers from purchasing the newer product.[166] Consumers struggle to learn the basics of computer operation and therefore seek uniformity in process and application.[167] They do not purchase software for its unique or aesthetic value.[168] Rather, consumers demand user-friendly products that maintain the same features and characteristics of previous

161. *See* Wilson, *supra* note 39, at 30.

162. *Id.*

163. *See id.* at 35.

164. *See generally id.* at 30–31 (noting that while functional similarity may legally exist, there may still be misappropriation).

165. *See infra* Part IV (discussing the realities of programmers changing employers and reinventing code to avoid copying).

166. *See* John C. Phillips, Note, *Sui Generis Intellectual Property Protection for Computer Software*, 60 GEO. WASH. L. REV. 997, 1009 (1992).

167. *See* Amin, *supra* note 31, at 34.

168. Phillips, *supra* note 166, at 1009.

versions at a lower cost.[169] In short, consumers want software that is inexpensive to purchase and easy to use.[170] Continued variation and upgrades increase learning curves discourages consumers and creates disincentives in the market.[171]

Reinvention, in the software context, is a model of inefficiency.[172] Duplicating programming efforts results in substantial costs and a failure to "incorporate the lessons of past mistakes and inefficiencies into future software designs and implementations."[173] All stages of software development, including the design, flowchart, code, debug, and documentation, suffer from these inefficiencies.[174] Companies also incur additional indirect costs in such things as providing technical support and fixing interoperability problems associated with new software.[175] Experts estimate the cost of this poor quality—or better stated, the cost of attempting to reproduce quality—between two billion and one hundred billion dollars annually.[176]

D. The Benefits of Reuse – A Better Model

If existing work is available, software companies can benefit by reusing and building upon existing foundations. The software reuse theory incorporates literal or non-literal pre-existing elements into a new program and encompasses outright copying of some or all of a computer program. It also may include utilization or application of comparable structure, sequence, and organization.[177]

The benefits of a reuse model far exceed any disadvantages. Reuse improves overall product quality as well as the quality of the detailed code

169. Amin, *supra* note 31, at 33. ("Diversity is the primary goal when it comes to novels, songs, and other traditional domains of copyright. Readers want to read novels they have not read. But diversity is not the goal of interface design. Computer users want consistency in interfaces because this promotes ease of use."). *Id.* at 33 n.105.

170. *See id.* at 33.

171. *See* Phillips, *supra* note 166, at 1009.

172. *See* Lemley, *supra* note 106, at 260.

173. *Id.*

174. *See id.*

175. *Id.*

176. *Id.*

177. *See* Risch, *supra* note 112, at 513.

because bugs are identified and fixed.[178] Moreover, reuse increases programmer efficiency and reduces the overall time to develop a new software package.[179] Lastly, reuse aids consumers by allowing them to build on the understanding and knowledge they have already acquired rather than forcing them to learn an entirely new program.[180]

Critics of the reuse theory advocate that disclosure is irrelevant to the fundamental premise of copyright because copyright evolves from creation, not registration.[181] However, the right established by creation is an imperfect right.[182] Companies register their code for the ability to enforce the right in the event of copyright infringement.[183] Thus, while registration is optional, to obtain the "panoply of remedies" available under the Copyright Act, the creator must register the work with the Copyright Office.[184]

Proponents of the reuse theory challenge computer programmers to follow more traditional scientific approaches.[185] For example, engineering disciplines routinely rely on widely known and developed theories and limitations and build upon existing components and structures.[186] By contrast, programmers incur high costs and encounter reliability and structural limitations as a result of forced reinvention.[187]

In fact, some advocates argue that the current method of protecting computer programs is inconsistent with the very function and purpose of the program itself.[188] They assert that a computer program is not a literary work at all, but rather a science continuously evolving for the betterment of

178. Lemley, *supra* note 106, at 265.

179. *Id.* at 265 n.66. Companies report between 40-57% increases in productivity. Some reports indicate an increase from 12.4 to 19 lines of code per day. *Id.*

180. *Id.* at 265.

181. *See generally* U.S. CONST. art. I, § 8, cl. 8 (granting copyright protection "to promote the Progress of Science and useful Arts").

182. *See generally* 17 U.S.C. § 411(a) (stating that the right is not perfected until the copyright is registered).

183. *See id.*

184. Schrader, *supra* note 130, at 126–27; *see also* 17 U.S.C. § 411.

185. *See* Lemley, *supra* note 106, at 256–57.

186. *See id.*

187. *See id.* at 257 (suggesting that software should be treated more as a scientific discipline than categorized as an art form).

188. *See* Stallman, *supra* note 23, at 2. *But cf.* Bobko, *supra* note 42, at 52–53.

its users.[189] If it is a science, programmers should distribute and share code, forcing replication, thereby creating efficiency.[190] One can compare the computer programming industry to the early days of the pharmaceutical industry wherein a small number of academic and medical practitioners generally conducted pharmaceutical research.[191] Today, an amalgamated industry drives molecular and medical technology, thereby propelling innovation forward.[192] Constant sharing of information and publication of processes, *via* patent protection and the disclosure requirements associated with obtaining such protections, allow for continued discovery and replication that makes science robust.[193]

Software development can apply a similar *modus operandi*. Disclosure of code, *via* full disclosure requirements, would provide public access to the computer code. Accessible code enables increased replication, which results in newer releases to programs that are more efficient. Additionally, this leads to continued creation or discovery of new software products.[194]

Lastly, as with many of today's technology driven processes, technological advancement in the application-programming field continues to systematize the manual programming process.[195] New programs that are designed by computers, rather than humans, will rely on existing programs and innovation to combine programs in a new and useful way.[196] Technological advances, coupled with the need for greater efficiency at reduced costs, will compel reevaluation of this protection quagmire.

V. OPEN SOURCE —A BETTER MODEL

Open source is a model of software creation and reuse that allows users access to all code in both source and object code forms.[197] The model

189. Stallman, *supra* note 23, at 7.

190. *Id.*

191. *See id.* at 4.

192. *See id.*

193. *See id.* at 7.

194. *See* Stallman, *supra* note 23, at 7.

195. *See* Jack M. Haynes, Comment, *Computer Software: Intellectual Property Protection in the United States and Japan*, 13 J. MARSHALL J. COMPUTER & INFO. L. 245, 258 (1995).

196. *See* Menell, *supra* note 65, at 1053–1055.

197. *See* McJohn, *supra* note 10, at 25.

thus promotes free access to computer code.[198] In fact, the free availability and malleability of computer code is the bedrock of the open source theory.[199] Examples of open source code products include Linux[200]and Apache.[201]

Non-profit groups, such as the Open Source Initiative ("OSI"), lead the cause for code disclosure.[202] OSI is an organization "dedicated to managing and promoting the Open Source Definition."[203] The precept of their position is that software evolves and science advances because all programmers read, modify, and redistribute source code freely.[204]

The open source model addresses the concern of high labor creation costs by selling the services that maintain and support the software produced rather than the software itself.[205] Programmers, in the open source model, are not highly compensated employees driven by corporate salaries, but rather are individual hackers[206] contributing to technology. Programmers' compensation and rewards are not monetary but instead are the prestige and recognition of contributing to and improving a body of knowledge.[207] Sheer enjoyment of programming and the desire to boast

198. *See* Andrew Leonard, *Who Controls Free Software,?* SALON.COM (Nov. 18, 1999), *available at* http://www.salon.com/tech/feature/1999/11/18/red_hat/print.html.

199. *See, e.g.*, The Apache Software Foundation, *at* http://www.apache.org/foundation/faq.html (last visited Aug. 21, 2003) (detailing the primary focus of the Apache Foundation as to "provide a foundation for open, collaborative, software development" where "individuals can donate resources . . . [that] . . . will be used for public benefit").

200. *See* The Linux Home Page at Linux Online, *at* http://www.linux.org (last visited Feb. 3, 2002).

201. *See* The Apache Software Foundation, *at* http://www.apache.org/foundation/faq.html (last visited Aug. 21, 2003).

202. *See, e.g.*, The Open Source Initiative: Home Page, *at* http://www.opensource.org (last visited Oct. 25, 2003).

203. *Id.*

204. *See id.* (detailing the basic premise of the organization as reading, redistributing and modifying source code to produce a superior product at a rapid pace). *See generally* Eric S. Raymond, *How to Become a Hacker*, *at*, http://tuxedo.org/~esr/faqs/hacker-howto.html (last visited Feb. 3, 2002) (describing the hacker attitude as believing in "freedom and voluntary mutual help").

205. *See* Bobko, *supra* note 42, at 82–84.

206. Hackers can further be defined as "a community, a shared culture, of expert programmers and networking wizards." *See generally* Raymond, *supra* note 203.

207. *See* Bobko, *supra* note 42, at 82–84. For example, an Australian team of hackers created and attached a cryptographic program to Netscape hours after its release. The

technical dexterity offer additional incentives.[208]

Advocates of the open source model champion a moral ideology.[209] "What we [advocates of the open source movement] all have in common is this desire to protect the interests of the public domain of knowledge."[210] The objective of this model is free software, defining freedom as "liberty" not "gratis."[211] In other words, "[f]ree as in speech, not as in beer."[212] This ideology is premised on the belief that every hacker faces a moral choice: self-enrichment which promotes betrayal by building walls that divide,[213] or societal enrichment which creates bridges that connect.[214]

The "free software" approach contradicts the established standards of copyright protection for computer software. In fact, some assert that the open source model capsizes traditional intellectual property laws that guard exclusive rights.[215] Commonly known as "copyleft,"[216] the open source model supports protection by preserving free software and manipulates current intellectual property laws to safeguard free access and use.[217]

---

cryptographic program enabled Netscape to conduct secured transactions on the Internet. *Id.* Although the team was not compensated by Netscape, their compensation came later in the form of recognition for their contribution and their ability to subsequently price for-profit projects accordingly. *Id.* at 83–84.

208. *See* McJohn, *supra* note 10, at 37 (discussing the non-monetary incentives of open source).

209. *See* Stallman, *supra* note 23, at 55.

210. *Ideological and Commercial Reasons for Open Source Were Aired and Fought over at Last Week's Code Conference*, THE GUARDIAN: WORLD REPORTER, Apr. 12, 2001, 2001 WL 18835392 (quoting Bob Young, the chairman of Red Hat, which distributes a well-known version of Linux).

211. Stallman, *supra* note 23, at 3.

212. *Id.* The mission of the open source movement has commonly been referred to by this phrase, to assist people in understanding its premise.

213. *Id.* (discussing Richard Stallman's decision to enter the free software endeavor).

214. *See id.*

215. *See id.* at 35, 67 (use of free software such as Gnome and Harmony make "non-free software" unneeded).

216. INTELLECTUAL PROPERTY ISSUES IN SOFTWARE, COMPUTER SCIENCE AND TELECOMMUNICATIONS BOARD 18 n.1 (National Academy Press 1991), *available at* http://books.nap.edu/books/0309043441/html/ [hereinafter "Computer Science"] (describing the Free Software Foundation's ("FSF") conviction against strong intellectual property laws for software). The FSF promotes copyright protection in the form of licensing agreements believing that innovation is best served when software is free. *Id.*; *see also* Stallman, *supra* note 23, at 59–60.

217. *See* Bobko, *supra* note 42, at 80–81.

Creators of computer programs prevent others from obtaining proprietary interests in their derivative works by means of a General Public License ("GPL").[218] The GPL is a tool utilized by the open source community which "perpetuates [its] particular software development and distribution model."[219] This licensing tool mirrors the idea that all improvements to the code must be made available in both source and object code form.[220]

Although this model contradicts the established standards of copyright protection, it achieves the fundamental goal of copyright by advancing the public good, while avoiding the divisive issue of classifying works as expressions or ideas in order to determine infringement.[221] The open source model allows programmers to develop new software or modify and enhance existing software, both of which benefit society while simultaneously creating a profit.[222] Yet, this model avoids the determination of who owns what monopoly, what is new, what is non-literal, what is structural or sequenced, and what can be filtered out leaving what can be determined as infringing.[223] In fact, it is virtually litigation-free provided users adhere to the licensing agreement.[224] Even if litigation does occur, the cause of action is often misappropriation, which is not as difficult or expensive to determine as copyright infringement of computer code.[225]

The open source model also eliminates the incentive to pirate software.[226] The volatile job market that entices programmers to move to

---

218. *Id.* For a more detailed description on various types of licenses available and provisions contained in those licenses, see McJohn, *supra* note 10, at 32–35.

219. Bobko, *supra* note 42, at 81. This licensing paragon operates similarly to the commercial software license whereby the licensee must agree not to distribute the software or establish proprietary rights in the software. *Id. See generally* Robert W. Gomulkiewicz & Mary L. Williamson, *A Brief Defense of Mass Market Software License Agreements*, 22 RUTGERS COMPUTER & TECH. L.J. 335 (1996) (providing a detailed discussion on mass marketing licenses).

220. *See* Bokbo, *supra* note 42, at 81.

221. *Id.* at 82–84.

222. *Id.*

223. *Id.* at 52–53.

224. *See generally* Stallman, *supra* note 23 (discussing various types of licensing schemes available from the numerous open source models).

225. *See* Wilson, *supra* note 38, at 39. "Copying of computer programs is easily disguised. Spotting a clever copy job is not easy for an expert and impossible for the layman, such as a judge, juror, or untutored lawyer." *Id.* at 34.

226. *See* Bobko, *supra* note 42, at 52 (discussing the differing economic incentives of the

the highest bidder fosters a transfer of knowledge, skill, and prior work product that must be avoided to avert infringement.[227] Subsequently, courts are left with the difficult task of determining what skills and knowledge a programmer transferred legitimately as individual expertise and what the employer can retain as proprietary information.[228] One court stated, "even in the best of good faith, a former technical employee working for a competitor, or in business for himself in a related field, can hardly prevent his knowledge of his former employer's confidential methods or data from showing up in his work."[229] By sharing code for the betterment of a common purpose, the incentive to pirate is virtually eliminated.

Another primary benefit of the open source model is the production of a superior product. Open code is accessible to an infinite number of processors, hackers, and computer hobbyists who are constantly changing, innovating, improving, and testing the current document.[230] Accordingly, the code is subject to rigorous and continuous peer review.[231] By contrast, for-profit companies operating primarily under a closed source model closely guard their code's secrecy.[232] Consequently, the closed source code is never subject to peer review because of the secrecy surrounding the code and the limited access to it.[233] The resulting product is therefore inferior and often unreliable.[234]

Timely and recurrent releases of open code contribute to its efficiency.[235] Continued releases allow thousands of hackers to work to eliminate bugs attributable to all computer programs.[236] Moreover, the

---

open source model).

227. *See infra* Part IV (discussing the flaws in the traditional model due to the programmers changing jobs).

228. *See* Wilson, *supra* note 39, at 30 (discussing the value of programmers as employees).

229. *See* Modern Controls, Inc. v. Andreadakis, 578 F.2d 1264, 1270 (8th Cir. 1978).

230. See Bobko, supra note 42, at 76 (citing Nikki Goth Itio, Freeware, Red Herring Mag. available at http://www.redherring.com/mag/issue63/news-freeware.html (Feb. 1999).

231. *Id.* at 77.

232. *See id.*

233. *Id.*

234. *See id.*

235. *See* Bobko, *supra* note 42, at 79 (by contrast, the closed source fixes software for long periods of time until an upgrade to the product is released by the same company that produced the original product).

236. *Id.* at 77–78. Bugs are defined as "problematic kinks" in a program's source code that contribute to its inefficiency. *Id.*

"fixes" or solutions, to these bugs are generally superior because the people that produce them do so voluntarily, and select the portions of the code where they are most proficient.[237] Users do not have to wait years for the next software release, which is typical of the proprietary model.[238] The constant reparation process ultimately benefits consumers who can download "fixes" to deficiencies in computer programs upon availability.[239]

Critics of the open source model maintain that the very premise of copyright is the reward to the creator.[240] However, while the immediate effect of copyright protection may be to procure a fair return to the creator for his efforts, the ultimate aim of providing protection is to promote and stimulate further creativity for the advancement of the public.[241] The Supreme Court, in *Feist Publications, Inc. v. Rural Telephone Service Co.*,[242] reinforced the notion that "the primary objective of copyright law is not to reward the labor of authors."[243] In *Feist*, Rural Telephone Service ("Rural") sued for copyright infringement seeking copyright protection for the labor required to compile a telephone white pages directory.[244] Rural asserted the "sweat of the brow" doctrine, which granted copyright protection to all compilations including facts thereby rewarding the efforts exerted in compiling primarily factual information.[245] The Court abrogated the "sweat of the brow" doctrine by denying copyright protection for a primarily factual compilation.[246] In rejecting *Feist's* rationale, the Supreme Court emphasized that the "sweat of the brow" doctrine "eschewed the

237. By contrast, in closed code fewer processors create, develop, and test the complete code. *Id.* at 78.

238. *Id.* at 79.

239. Microsoft, on the other hand, maintains its market power by its continued "iterations" of software that must be purchased. *Id.* at 78–79.

240. *See id.* at 51.

241. Bobko, *supra* note 42, at 74.

242. 499 U.S. 340 (1991).

243. *Id.* at 349.

244. *Id.* at 340.

245. *Id.* at 352; *see also* Computer Assocs. Int'l, Inc. v. Atai, Inc., 982 F.2d 693, 712 (2d Cir. 1992) (asserting that the same rationale should be applied to computer code which is also primarily factual).

246. *Feist Publ'ns.*, 499 U.S. at 353. *But see* CCC Info. Servs., Inc. v. Maclean Hunter Mkt. Reports, Inc., 44 F.3d 61 (2d Cir. 1994) (finding copyright protection for compilation of car valuations); *see also* Kregos v. Associated Press, 937 F.2d 700, 710 (2d Cir. 1991) (holding that defeating a claim for infringement of a compilation of factual information requires only a showing that the differences in the works are "more than trivial").

most fundamental axiom of copyright law that no one may copyright facts or ideas."[247] Thus, the Court proclaimed that the primary objective of copyright law was not to reward creators for their efforts.[248]

Moreover, tipping the scales too far leads to overprotection while simultaneously resulting in a depletion of valuable basic ingredients. Authors do not invent every component of what they create. Instead, they rely on ideas, common themes, and stock phrases.[249] Providing copyright protection for this information unduly burdens the creative arts by depleting society of the basic building blocks needed to construct future foundations.[250] This over-exhaustive protection may deter authors from producing anything new. Those that did would have to expend an exorbitant amount of time and money, but their final product may still lack luster. Thus the "imposition of limits must be seen as a vital and integral part of copyright's structural function."[251] Open source is the progressive solution that satisfies the goals of copyright while simultaneously advancing society.

The open source ideology continues to gain popularity as a viable alternative to the traditional closed source model. Industry giants such as IBM adopted the Linux system and have promoted its benefits enthusiastically.[252] Additionally, companies such as L.L.Bean have also begun the conversion.[253] Netscape and Intel invested in the Linux-distributed software known as Red Hat.[254] These companies are building and supporting a foundation that will benefit them specifically, while contributing to science and technology to benefit society generally.[255]

---

247. *Feist Publ'ns.*, 499 U.S. at 341.

248. *Id.* at 349 (nullifying the "sweat of the brow" doctrine); *see also supra* Part III.

249. *See* Nichols v. Universal Pictures Corp., 45 F.2d 119, 121 (2d Cir. 1930), *cert. denied* 282 U.S. 902 (stating that common themes and stock figures are not copyrightable).

250. *See* Neil Weinstock Netanel, *Copyright and a Democratic Civil Society*, 106 YALE L.J. 283, 362 (1996).

251. *Id.*

252. *See generally* Stephen Shankland, *IBM: Linux Can Take on the World*, ZDNET NEWS (Jan. 31, 2002), *at* http://zdnet.com.com/2100-1106-826983.html (touting IBM's adoption and promotion of the Linux system).

253. *See generally id.* (discussing L.L.Bean's recent adoption of several components of Linux and the superior performance without fail).

254. Stallman, *supra* note 23, at 186.

255. *See id.* at 187–88.

VI. CONCLUSION

The development of computer software is the most rapidly expanding and most profitable business model in today's high tech computer world.[256] It has extreme value in today's society, both economically and functionally. Legal protection for software should provide an incentive structure that leads to continuous creation and innovation. The current alignment of protection under copyright law does not provide this incentive, but rather discourages the dissemination of new inventions because it encourages reinvention to circumvent copyright infringement. As one theorist has postured, "[w]e're trying to use 19th century legal tools to accommodate 21st century information technology."[257]

Moreover, courts continue to struggle with interpreting and applying copyright law to computer programs [258]in an attempt to treat such programs as analogous to literary works.[259] "Judges have been blind to the fact that software is a technology and that progress in the field of technological arts may more easily be impeded by strong copyright protection than might be the case in the field of literary arts."[260]

Computer program protection, under the present *modus operandi*, collides with the fundamental principles of copyright. Traditional copyright principles reward creators by providing protection for their work, while simultaneously creating an economic incentive and a foundation for other creators to build upon.[261] Indeed, the very foundation of copyright law founded in the Constitution is "[t]o promote the Progress of Science and the useful Arts by securing for limited Times to Authors and Inventors the exclusive right to their respective Writings and Discoveries."[262] Although the Supreme Court advocates that the primary focus of copyright is not to reward the creator by providing him a monopoly,[263] the current process of

256. *See id.* at 195.

257. Bencivenga, *supra* note 144 (quoting Joel R. Reidenberg, associate professor at Fordham University School of Law who specializes in information technology).

258. Computer Associates Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 696 (2d Cir. 1992).

259. COMPUTER SCIENCE, *supra* note 215, at 29.

260. *Id.* (quoting Professor Pamela Samuelson, Professor of Law, University of Pittsburgh).

261. *See* Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 711 (2d Cir. 1992).

262. U.S. CONST. art. I, § 8, cl. 8.

263. *See* Feist Publ'ns., Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 354 (1991) (quoting MELVILLE B. NIMMER & DAVID NIMMER, 1-3 NIMMER ON COPYRIGHT § 3.04, p. 3-23 (2003);

protection via the 1976 Act (as amended) adopted by Congress seems to support that intent. By enabling a software developer to block out code to prevent others from building on or improving such code while simultaneously obtaining copyright protection, Congress encourages the acquisition of wealth, which benefits few and discourages the advancement of society that will benefit many.[264] Moreover, blocked-out protection forces developers to constantly reinvent the wheel rather than reuse the innovation.[265]

Open source obviates this paradigm. As one advocate aptly stated [i]f we are ever going to lick this software crisis, we're going to have to stop this hand-to-mouth, every-programmer-builds-everything-from-the-ground-up, pre-industrial approach."[266] Open source programming relies on a reuse theory that is reliable, cheaper, and more efficient.[267] Because open source trumps commercially secret code in reliability, efficiency, and availability, Congress should embrace and encourage open source as the method of submission for copyright protection.

---

see also Twentieth Century Music Corp. v. Aiken, 422 U.S. 151, 156. (1975) (stating that the primary goal is to "promote broad public availability of literature, music and the other arts"). Because it is publicly available, open source achieves the goal while the traditional model contradicts this purpose. *See supra* Part V.

264. *See* Phillips, *supra* note 165, at 1004.

265. *See* Lemley, *supra* note 105, at 259–68.

266. *See id.* at 256.

267. *See id.*