

***BOWERS v. BAYSTATE TECHNOLOGIES: THE UNFAIR USE  
DOCTRINE OF REVERSE ENGINEERING***

*Nicholas Camillo\**

I.	INTRODUCTION .....	159
II.	COMPUTER SOFTWARE AND REVERSE ENGINEERING.....	160
III.	FEDERAL COPYRIGHT LAW AND ITS ATTEMPT TO PROTECT COMPUTER SOFTWARE .....	1633
IV.	COPYRIGHT FAIR USE: THE DEFENSE OF REVERSE ENGINEERING .....	167
V.	<i>BOWERS v. BAYSTATE TECHNOLOGIES, INC.</i> .....	172
VI.	THE INADEQUATE SOLUTION: FAIR USE .....	175
VII.	CONCLUSION.....	181

**I. INTRODUCTION**

The goal of copyright law is utilitarian in nature; it is to disseminate artistic and scientific works that will ultimately benefit society.<sup>1</sup> Copyright law provides an incentive for authors to create new expressive works by granting them exclusive rights so that they may receive economic benefits for their work.<sup>2</sup> Computer software has become difficult to place within the context of copyright law because it is challenging for authors of computer software to disseminate their ideas and simultaneously control access to their work so that they may benefit financially from its reproduction. The fair use defense for reverse

---

\* J.D., University of Florida Levin College of Law (2016); President, Intellectual Property and Technology Law Association (2015-2016); Research Assistant to Professor Elizabeth Rowe (2015-2016); Notes Editor, *Journal of Technology Law and Policy* (2015). I would like to thank Professor Richard Vermut for his constructive advice on earlier drafts, and Professor Elizabeth Rowe for her invaluable support and guidance in the field of intellectual property law. I would further like to thank the entire staff of the *Journal of Technology Law and Policy* for all of their hard work and effort.

1. U.S. CONST. art. I, § 8, cl. 8 (stating that copyrights “promote the Progress of Science and useful Arts”).

2. 17 U.S.C. § 106 (1976) (granting the copyright owner exclusive rights).

engineering is thought to be a positive limitation on copyright law that allows the public to use and understand ideas within a program, and therefore progress science and the useful arts.<sup>3</sup> *Bowers v. Baystate Technologies, Inc.* displays how courts tow a dangerous line when trying to not only protect the author from the financial hardship that results from the unauthorized appropriation of a program's code, but also prevent the author from having a monopoly over the ideas contained in the program.<sup>4</sup> This Note argues that reverse engineering should not constitute a copyright fair use defense and uses *Bowers* as a conduit to show why reverse engineering is an inadequate solution to the complications surrounding copyright law's protection of computer software.

Following this introduction, Part II of this Note will examine the use of reverse engineering to facilitate computer software innovation. Part III will discuss the permissible exclusive rights outlined in copyright law, how and why computer software is copyrightable, and how reverse engineering may be an infringement to copyright law. Part IV will analyze the fair use provision of the Copyright Act. Part V will describe the fundamental facts and opinions of *Bowers*. Finally, Part VI will use *Bowers* to explain why reverse engineering should not constitute a copyright fair use defense.

## II. COMPUTER SOFTWARE AND REVERSE ENGINEERING

As we have progressed through the twenty-first century, computer software has grown exponentially in terms of development and innovation.<sup>5</sup> Computer software works in conjunction with computer hardware to provide computers with applications and programs that allow an individual to accomplish various tasks on a computer.<sup>6</sup> Without computer software, a computer would be virtually useless.<sup>7</sup> There are generally two categories of computer software: system and application.<sup>8</sup> System software is used to operate the computer and is known as the

---

3. *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317, 1337 (3d Cir. 2003).

4. *Id.* at 1317.

5. See generally *Software Industry Facts and Figures*, Bus. Software Alliance, [http://www.bsa.org/country/Public%20Policy/~media/Files/Policy/Security/General/sw\\_factsfigures.ashx](http://www.bsa.org/country/Public%20Policy/~media/Files/Policy/Security/General/sw_factsfigures.ashx) (last visited May 5, 2015) (showing various charts and graphs describing the growth of the computer software industry).

6. Throughout this Note the terms: "program," "computer program," "software," and "computer software" are all used interchangeably. See, e.g., Daniel J. Smith, *Proof of Copyright Infringement by Unauthorized Use of Software*, 52 AM. JUR. 3D *Proof of Facts* 107 (1999) (providing a general overview of computer software).

7. *Id.*

8. *Id.*

“operating system.”<sup>9</sup> Well known operating systems include, among others, Apple Mac OS and Microsoft Windows.<sup>10</sup> Application software works with the operating system to perform specific tasks.<sup>11</sup>

All computer software is written with a computer language called “code.”<sup>12</sup> There are two types of code: object code and source code. Object code is a binary language composed of zeroes and ones that the computer receives its instructions from the software.<sup>13</sup> Source code is translated into object code when a program is ran so that the computer can read the program’s instructions.<sup>14</sup> There are various types of source code languages that are used depending on what type of software a developer is trying to create.<sup>15</sup> What an individual sees on the screen when using a program is a compilation of object code and source code working in conjunction to create an interface comprised of words and graphics that allows the individual to operate the program’s functions to complete a task.<sup>16</sup>

A key takeaway when noting the difference between object code and source code is that object code is represented in a language that only a computer can read, whereas source code is represented in a language that humans can read.<sup>17</sup> When a program is developed using source code, that program can be written in many different ways and still produce the same end function.<sup>18</sup> The syntax of a program’s source code, or the unique way that a program is written, speaks to the distinctiveness of the program.<sup>19</sup> If a software developer wants to understand the functionality of a program and what makes it distinctive from other similar programs, he or she can attempt to reverse engineer the program to extract the source code from the object code.<sup>20</sup>

To reverse engineer in the computer software context, an individual works backwards from object code to produce source code.<sup>21</sup> Reverse engineering ordinarily means, “to study or analyze (a device, as a microchip for computers) in order to learn details of design, construction,

---

9. *Id.*

10. *Id.*

11. *Id.*

12. *Id.*

13. *Id.*

14. *Id.*

15. *Id.*

16. *Id.*

17. See, e.g., Robert Lech, *Protecting Computer Software Against Reverse Engineering*, 73 MICH. B.J. 526-27 (1994).

18. *Id.*

19. *Id.*

20. *Id.*

21. *Id.*

and operation, perhaps to produce a copy or improved version.”<sup>22</sup> An individual who acquires computer software in a valid transaction can reverse engineer its source code to discern the composition of the program.<sup>23</sup> Once the program’s composition is understood, the individual can create a new program that is interoperable with the reverse engineered program, or a derivative program that competes with the original.<sup>24</sup>

Many individuals have acknowledged the positive impact that reverse engineering has on the computer software industry.<sup>25</sup> In fact, the U.S. Supreme Court has held that reverse engineering is an “essential part of innovation.”<sup>26</sup> However, this does not necessarily mean that software developers are interested in sharing the inner workings of their programs.<sup>27</sup> Commercial software developers generally distribute their software in object code form so that they can protect their source code as a trade secret, but this tactic does not guarantee protection.<sup>28</sup> Under the Uniform Trade Secrets Act, reverse engineering is a proper means of obtaining source code if the software is owned and was acquired on the open market, and the reverse engineering does not violate a confidential relationship.<sup>29</sup> Some people believe that reverse engineering should be illegal as a matter of copyright and trade secrecy law.<sup>30</sup> They believe that unauthorized copies of programs that are made via reverse engineering infringes the copyright owner’s exclusive right to reproduce the program in copies, which in turn makes reverse engineering an inappropriate process to obtain a program’s source code.<sup>31</sup>

---

22. *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317, 1326 (3d Cir. 2003) (quoting RANDOM HOUSE UNABRIDGED DICTIONARY (2d ed. 1993)).

23. *See* 10 Hawklund UCC Series UCITA § 118:1.

24. *See, e.g.*, Pamela Samuelson & Suzanne Scotchmer, *The Law and Economics of Reverse Engineering*, 111 YALE L.J. 1575, 1608 (2002).

25. *Id.*

26. Robert W. Gomulkiewicz, *Fostering the Business of Innovation: The Untold Story of Bowers v. Baystate Technologies*, 7 WASH. J.L. TECH. & ARTS 445, 449 (2012) (quoting *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, 489 U.S. 141, 160 (1989)).

27. *See, e.g.*, Samuelson & Scotchmer, *supra* note 24, at 1609.

28. *Id.* at 1608.

29. *See* Unif. Trade Secrets Act § 1 (1985) (defining improper means as “theft, bribery, misrepresentation, breach or inducement of a breach of a duty to maintain secrecy, or espionage through electronic or other means”).

30. *See, e.g.*, Samuelson & Scotchmer, *supra* note 24, at 1609.

31. *Id.*; *see also* 17 U.S.C. § 106(1) (stating that the owner of a copyright has the exclusive right to do and to authorize the reproduction of the work in copies or phonorecords).

### III. FEDERAL COPYRIGHT LAW AND ITS ATTEMPT TO PROTECT COMPUTER SOFTWARE

Current copyright laws attempt to protect the developers of computer software, whereas earlier laws did not make it clear as to what protections computer software should be afforded.<sup>32</sup> The Copyright Act states that “literary works” and other “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,” are afforded copyright protection.<sup>33</sup> In the seminal 1879 copyright case *Baker v. Seldon*, the court established the distinction between ideas and expression. In *Baker*, Seldon created a book that contained drawings and explanations for a unique system of bookkeeping.<sup>34</sup> Baker used Seldon’s system, and Seldon sued Baker for copyright infringement.<sup>35</sup> The Court held that “[t]he novelty of the art or thing described or explained has nothing to do with the validity of the copyright.”<sup>36</sup> The Court reasoned that granting Seldon an exclusive right over his bookkeeping idea would be a patent right, and not a right granted by copyright law.<sup>37</sup> The rule established in *Baker* has since been codified in Section 102(b) of the Copyright Act.<sup>38</sup>

Copyright law aims to protect expression over functionality, but computer software is unique in that it is a creation that incorporates both functionality and expression.<sup>39</sup> The National Commission on New Technological Uses of Copyrighted Works (CONTU) recommended to Congress that copyright laws offered the best protection to computer software.<sup>40</sup> In 1980, Congress amended the 1976 Copyright Act to include a definition of “computer program.”<sup>41</sup> According to Section 101 of the Copyright Act, “a ‘computer program’ is a set of statements or instructions to be used directly or indirectly in a computer in order to

---

32. See, e.g., Stacey L. Dogan & Joseph P. Liu, *Copyright Law and Subject Matter Specificity: The Case of Computer Software*, 61 N.Y.U. ANN. SURV. AM. L. 203, 207 (2005).

33. 17 U.S.C. § 102 (1976).

34. *Baker v. Seldon*, 101 U.S. 99 (1879).

35. *Id.*

36. *Id.* at 102.

37. *Id.*

38. 17 U.S.C. 102(b) (“In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work”).

39. See, e.g., Dogan & Liu, *supra* note 32, at 207.

40. *Id.* at 208.

41. 17 U.S.C. §§ 101, 117(a) (2000).

bring about a certain result.”<sup>42</sup> Congress left it up to the courts to decide what aspects of computer programs warranted protection and against what kinds of copying.<sup>43</sup>

Computer software falls under the protection of copyright law because copyright law protects expression rather than function, and software is written in a “language” consisting of symbols, letters, and numbers that we associate with literary works.<sup>44</sup> Accordingly, the Third Circuit confirmed that computer programs are in fact literary works protected under the Copyright Act in *Apple Computer, Inc. v. Franklin Computer Corp.*<sup>45</sup> In *Apple*, Franklin developed and sold an ACE 100 personal computer that was designed to be compatible with Apple’s peripheral equipment and software used in Apple’s Apple II computers.<sup>46</sup> Apple sued Franklin because he copied Apple’s operating system to achieve compatibility.<sup>47</sup> The court explained that “the definition of ‘literary works’ in section 101 include[s] expression not only in words but also ‘numbers, or other . . . numerical symbols or indicia,’ thereby expanding the common usage of ‘literary works.’”<sup>48</sup> Therefore, the court held that a computer program, “whether in object code or source code, is a ‘literary work’ and is protected from unauthorized copying, whether from its object or source code version.”<sup>49</sup> However, the court failed to consider the commercial and competitive intent of copying computer software and reasoned that the “total compatibility with independently developed application programs written for the Apple II” does not bear weight to the issue of whether ideas and expressions have merged, and thus has no relevance to copyright law.<sup>50</sup>

The court in *Apple* left a void in its reasoning when it came to understanding whether a program’s function could be protected by a copyright holder’s exclusive rights. In *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, the Third Circuit Court of Appeals considered whether a program’s function could be afforded protection by copyright law.<sup>51</sup> In *Whelan*, Jaslow Dental Laboratory (Jaslow Lab) hired Whelan Associates (Whelan) to create a program called Dentalab that could help Jaslow Lab’s dental business. Eventually, Jaslow Lab and Whelan came to an agreement to sell the Dentalab program to other members of the

---

42. 17 U.S.C. § 101 (1980).

43. Dogan & Liu, *supra* note 32, at 208.

44. *Id.*

45. *Apple Computers, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240 (3d Cir. 1983).

46. *Id.* at 1243.

47. *Id.*

48. *Id.* at 1249 (quoting *Harcourt, Brace & World, Inc. v. Graphic Controls Corp.*, 329 F. Supp. 517, 523-24 (S.D.N.Y. 1971)).

49. *Id.*

50. *Id.* at 1253.

51. *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222 (3d Cir. 1986).

dental industry.<sup>52</sup> Jaslow Lab eventually felt that they could create a similar program in a more basic coding language, and subsequently created a program called Dentcom and sold it to customers within the dental industry.<sup>53</sup> Jaslow Lab sent a letter to Whelan telling Whelan to stop selling the Dentalab program because it contained trade secrets of Jaslow Lab.<sup>54</sup> Whelan continued to sell the Dentalab program and litigation ensued.<sup>55</sup> Whelan claimed that Jaslow Lab had infringed Whelan's copyright in Dentalab.<sup>56</sup> Jaslow Lab argued that the computer program was an idea, not the expression of an idea, and thus cannot be protected by copyright.<sup>57</sup> The court held that the "copyright protection of computer programs may extend beyond the programs' literal code to their structure, sequence, and organization."<sup>58</sup> The court reasoned that

the line between idea and expression may be drawn with reference to the end sought to be achieved by the work in question . . . . [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 the idea.

The reasoning in *Whelan* drew criticism and the court in *Computer Associates International, Inc. v. Altai, Inc.* specifically distinguished itself from *Whelan*.<sup>59</sup>

In *Altai*, Computer Associates (CA) created a program called ADAPTER that allowed a computer user to change operating systems while maintaining the same software.<sup>60</sup> Claude Arney, an employee of Altai who previously worked for CA, created a program with a similar function called OSCAR that used 30% of the ADAPTER program's code.<sup>61</sup> CA sued Altai for copyright infringement.<sup>62</sup> The court believed that decisions involving copyright protection of computer software should consider context:

We think that *Whelan's* approach to separating idea from expression in computer programs relies too heavily on metaphysical distinctions and does not place enough emphasis on

---

52. *Id.* at 1225.

53. *Id.*

54. *Id.*

55. *Id.*

56. *Id.*

57. *Id.* at 1235.

58. *Id.* at 1248.

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

60. *Id.* at 669.

61. *Id.*

62. *Id.*

practical considerations . . . . As the cases that we shall discuss demonstrate, a satisfactory answer to this problem cannot be reached by resorting, a priori, to philosophical first principles.<sup>63</sup>

The court in *Altai* adopted a fact specific abstraction, filtration, and comparison approach.<sup>64</sup> In the abstraction step, the court looked to break down the program into its structural parts.<sup>65</sup> In the filtration step, the court examined the broken down program and sifted out of the inquiry any portions that were not protected by copyright law.<sup>66</sup> Finally, in the comparison step, the court compared the protectable material with the structure of the allegedly infringing program to see if the elements were substantially similar and thus warranted a finding of infringement.<sup>67</sup> The court found that efficiency, with respect to computer software, is an industry-wide goal, and that there may only be a certain number of ways to create the same function within a program.<sup>68</sup> Thus, the idea merged with its expression,<sup>69</sup> and the court held, “(1) that programmers may receive appropriate copyright protection for innovative utilitarian works containing expression; and (2) that non-protectable technical expression remains in the public domain for others to use freely as building blocks in their own work.”<sup>70</sup> The court drew upon policy considerations to confirm its application of the three-step approach, stating that

any method that tries to distinguish idea from expression ultimately impacts on the scope of copyright protection afforded to a particular type of work . . . , [and] [t]he interest of the copyright law is not in simply conferring a monopoly on industrious persons, but in advancing the public welfare through rewarding artistic creativity, in a manner that permits the free use and development of non-protectable ideas and processes.<sup>71</sup>

*Apple*, *Whelan*, and *Altai* still leave individuals with a hazy interpretation as to what parts of a program are protected by copyright

---

63. *Id.* at 706; *see also* Dogan & Liu, *supra* note 32, at 207 (stating that *Altai* moved away from the detached approach used by earlier software cases, and moved towards considering software cases in context).

64. *Id.*

65. *Id.*

66. *Id.* (“by examining each of these parts for such things as incorporated ideas, expression that is necessarily incidental to those ideas, and elements that are taken from the public domain, a court would then be able to sift out all non-protectable material”).

67. *Id.*

68. *Id.* at 708.

69. *Id.*

70. *Id.* at 721.

71. *Id.* at 711.

law. With each court's interpretation, a clear distinction between idea and expression seems to only become more convoluted. But, what is generally understood is that the Copyright Act grants, with certain exceptions, exclusive rights to use and to authorize use of an owner's copyrighted work. Generally, the owner of a copyrighted work has the exclusive rights to: (1) reproduce the copyrighted work in copies; (2) prepare derivative works; (3) distribute copies by sale, transfer of ownership, rental, lease, or lending; (4) perform the copyrighted work publicly; (5) display the copyright work publicly; and (6) perform a digital audio transmission of the copyrighted work publicly.<sup>72</sup> The protection afforded to a copyright owner by the Copyright Act does not provide an owner with complete control over all uses of an individual's work.<sup>73</sup>

The statutory limitations placed on a copyright holder's exclusive rights and the lack of defined protection afforded to the function of a program, means that reverse engineering poses a legitimate threat to computer software developers. The Digital Millennium Copyright Act (DMCA), enacted in 1998, imposes liability on any person that circumvents a technological measure that controls access to a copyrighted work, and prohibits the distribution of technology that facilitates circumvention.<sup>74</sup> However, reverse engineering for the purpose of analyzing elements of a program to achieve interoperability of an independently created program with other programs is permitted under the DMCA.<sup>75</sup> Whether reverse engineering copyrightable computer software code is legal is contingent on whether making copies to analyze the code infringes the software copyright.<sup>76</sup> An individual making a copy of the code conflicts with the copyright owners exclusive right to make copies, but the legality of the copying turns on whether the making of the copies is protected under the fair use doctrine of the Copyright Act.<sup>77</sup>

#### IV. COPYRIGHT FAIR USE: THE DEFENSE OF REVERSE ENGINEERING

The fair use defense in copyright law permits certain exceptions to copyright infringement. The goal of the fair use defense is to further "promote the Progress of Science and useful Arts" by permitting certain

---

72. 17 U.S.C. § 106 (1976). Parts 4 and 5 of the exclusive rights section of the Copyright Act are with respect to "literary, musical, dramatic, and choreographic works, pantomimes, and motion pictures and other audiovisual works" while part 6 is with respect to sound recordings.

73. See Lisa A. Zakolski, *Copyright and Literary Property*, 18 AM. JUR. 2D *Copyright & Literary Property* § 69 (2015).

74. 17 U.S.C. § 1201 (1998).

75. *Id.* at (f).

76. See Raymond Nimmer, *Reverse Engineering in General*, 10 Hawklnd UCC Series UCITA § 118:1 (last modified Jan. 2015).

77. *Id.*

activities that would generally constitute copyright infringement.<sup>78</sup> According to Section 107 of the Copyright Act, it is not copyright infringement to make copies of a copyrighted work “for purposes such as criticism, comment, news reporting, teaching (including multiple copies for classroom use), scholarship, or research . . . .”<sup>79</sup> The factors considered in determining whether the use made of a copyrighted work is a fair use are:

- (1) the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes;
- (2) the nature of the copyrighted work;
- (3) the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
- (4) the effect of the use upon the potential market for or value of the copyrighted work.<sup>80</sup>

The factors are intended to be guidelines and to provide consistency when determining the application of the fair use defense. In the early 1990s, several courts expanded the scope of the fair use defense to permit the intermediate copying of software for the purpose of reverse engineering.<sup>81</sup> Fair use is often argued when a defendant reverse engineers a copyrighted computer program to gain access to the unprotectable source code by copying the protected object code.<sup>82</sup>

The first decision that tested the application of fair use to the reverse engineering of computer software was *Sega Enters., Ltd. v. Accolade, Inc.*<sup>83</sup> In that case, Sega Enterprises was a Japanese corporation that developed and marketed its own video game systems and cartridges, including its own “Genesis” console.<sup>84</sup> Accolade was a developer, manufacturer, and marketer of video games, including games that were compatible with the Genesis console.<sup>85</sup> Accolade planned on becoming a licensee of Sega to develop and sell Genesis-compatible games, but refrained from doing so because Sega demanded that it be the exclusive manufacturer of any games that Accolade produced.<sup>86</sup> Accolade instead

---

78. U.S. CONST. art. I § 8, cl. 8; see Karen E. Georgenson, *Reverse Engineering of Copyrighted Software: Fair Use or Misuse?*, 5 ALB. L.J. SCI. & TECH. 291 (1996).

79. 17 U.S.C. § 107.

80. *Id.*

81. See Robert W. Gomulkiewicz, *Fostering the Business of Innovation: The Untold Story of Bowers v. Baystate Technologies*, 7 WASH. J.L. TECH. & ARTS 445 (2012); Georgenson, *supra* note 78, at 291.

82. Elga A. Goodman et al., *Defenses to Copyright Infringement-Fair Use of Computer Programs*, 49 N.J. PRAC. BUSINESS LAW DESKBOOK § 11:17 (2014).

83. *Sega Enter. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992).

84. *Id.* at 1514.

85. *Id.*

86. *Id.*

reverse engineered Sega's video game programs, studied the interaction between the video game programs and the Genesis console, and created a development manual that contained their findings.<sup>87</sup> The development manual did not contain any of Sega's code.<sup>88</sup> It only contained functional descriptions of the interface requirements for the Genesis console.<sup>89</sup> Subsequently, Accolade created its own games for the Genesis console without copying any of Sega's programs, relying only on the information contained in the development manual.<sup>90</sup>

Sega eventually created a new Genesis console, the "Genesis III," that contained an initialization code that prompts a game to display "PRODUCED BY OR UNDER LICENSE FROM SEGA ENTERPRISES LTD" on start up.<sup>91</sup> When the new Genesis III was released, Accolade reverse engineered the Genesis III console to understand the initialization code.<sup>92</sup> Accolade added the initialization code to its development manual, and future released games adopted the code. The initialization code was the only portion of Sega's code that Accolade copied into its video game programs.<sup>93</sup> Sega filed a suit against Accolade for copyright infringement and both parties filed cross-motions to enjoin the other. Accolade argued that the intermediate copying of Sega's game programs was fair use, and thus did not constitute copyright infringement.<sup>94</sup>

The court in *Sega* applied the four factors test of fair use, outlined in Section 107 of the Copyright Act, to Accolade's reverse engineering of Sega's game programs. The court ruled in favor of Accolade with respect to the first statutory factor, stating that Accolade's "direct purpose in copying Sega's code, and thus its direct use of the copyrighted material, was simply to study the functional requirements for Genesis compatibility so that it could modify existing games and make them usable with the Genesis console."<sup>95</sup> The court determined that the second factor, nature of the copyrighted work, granted Sega a lower degree of protection than other literary works because the "disassembly of the object code in Sega's video game cartridges was necessary in order to understand the functional requirements for Genesis compatibility," and thus weighed in Accolade's favor.<sup>96</sup> Regarding the third factor, the court

---

87. *Id.* at 1515.

88. *Id.*

89. *Id.*

90. *Id.*

91. *Id.*

92. *Id.* at 1516.

93. *Id.* (the initialization code contained around 20 to 25 bytes of data while Accolade's games contained 500,000 to 1,500,000 bytes of data).

94. *Id.*

95. *Id.* at 1522.

96. *Id.* at 1526.

negatively weighed it against Accolade, but noted that the factor carried little influence considering that fair use is not precluded even when a work is copied, especially when the ultimate use is limited.<sup>97</sup> Finally, the court considered the fourth factor and found that Accolade's reverse engineering of Sega's software did not significantly affect the market for Sega's games and noted that it was unlawful for Sega to have a de facto monopoly over the market by making it impossible for others to make competing products.<sup>98</sup> The court concluded "that where disassembly is the only way to gain access to the ideas and functional elements embodied in a copyrighted computer program and where there is a legitimate reason for seeking such access, disassembly is a fair use of the copyrighted work, as a matter of law."<sup>99</sup>

Another case that affirmed the applicability of the fair use defense to the reverse engineering of computer software was *Atari Games Corp. v. Nintendo of America, Inc.*<sup>100</sup> In *Atari*, Nintendo created a chip with an embedded program called 10NES that would prevent the NES game console from accepting unauthorized game cartridges.<sup>101</sup> Atari tried multiple times to analyze and replicate the 10NES program but failed.<sup>102</sup> In 1987, Atari became a licensee of Nintendo, which allowed them to make games for the NES console, but contained terms that controlled Atari's access to the NES technology, including the 10NES program.<sup>103</sup> In 1988, Atari's attorney lied to the Copyright Office, stating that Atari was a defendant in an infringement case and needed a copy of the 10NES program for litigation.<sup>104</sup> After acquiring the 10NES source code from the Copyright Office, Atari replicated the 10NES program and developed its own program called the Rabbit program to gain access to NES owners.<sup>105</sup>

Nintendo filed a motion to enjoin Atari and prevailed.<sup>106</sup> Atari subsequently appealed.<sup>107</sup> The court found that Section 107 of the Copyright Act allowed an individual to possess an authorized copy of a work to understand the work's ideas, processes, and methods of operation.<sup>108</sup> The court held that "[w]hen the nature of a work requires intermediate copying to understand the ideas and processes in a

---

97. *Id.* at 1526–27.

98. *Id.* at 1523.

99. *Id.* at 1527.

100. *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832 (Fed. Cir. 1992).

101. *Id.* at 836

102. *Id.*

103. *Id.*

104. *Id.*

105. *Id.* at 837.

106. *Id.*

107. *Id.*

108. *Id.* at 842.

copyrighted work, that nature supports a fair use for intermediate copying. Thus, reverse engineering object code to discern the unprotectable ideas in a computer program is a fair use.”<sup>109</sup> However, the court did not extend the fair use defense to Atari because “[f]air use to discern a work’s ideas, however, does not justify extensive efforts to profit from replicating protected expression. Subparagraphs 1 and 4 of section 107 clarify that the fair use in intermediate copying does not extend to commercial exploitation of protected expression.”<sup>110</sup>

Both of the courts in *Sega* and *Atari* ruled that the fair use defense could be applied to the reverse engineering of computer software, but that does not mean that copyright law has completely been adapted to accommodate computer software. Although the courts in both *Sega* and *Atari* confirmed the legality of reverse engineering, there is a lack of uniformity in analysis when applying the fair use defense.<sup>111</sup> Furthermore, Section 117 of the Copyright Act adds to the lack of uniformity by permitting the owner of a copy of a computer program to copy or adapt the program provided that the copy or adaptation is “an essential step in the utilization of the computer program in conjunction with a machine and . . . is used in no other manner.”<sup>112</sup> The court in *Sega* saw that Section 117 defined a narrow category of copying that was lawful *per se*, and Section 107 established a defense to a copyright infringement claim.<sup>113</sup>

Because individuals can circumvent a developer’s copyright protection by reverse engineering the object code into source code, most software developers started to require users to agree contractually that they would not reverse engineer the object code to discover the software’s source code.<sup>114</sup> This shift to contract law is where *Bowers v. Baystate Technologies, Inc.* finds its place amongst the convoluted adaptation of the fair use defense for reverse engineering.<sup>115</sup> By enforcing a license agreement whose terms directly conflicted with established copyright law, *Bowers* set a precedent that ultimately proves that copyright law needs to be more transparent in the context of computer software.<sup>116</sup>

---

109. *Id.* at 843.

110. *Id.*

111. See Georgenson, *supra* note 78, at 308 (citing *Atari*, 975 F.2d at 832; *Sega Enters., Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992) (explaining that *Atari* focused on the need to understand copyrighted material, while *Sega* applied the statutory factors in greater detail than the court in *Atari*, and emphasized the importance of public policy when determining if the fair use doctrine is applicable)).

112. *Sega*, 977 F.2d at 1510 (quoting 17 U.S.C. § 117(a)(1)).

113. *Id.* at 1521.

114. David Pruitt, *Beyond Fair Use: The Right to Contract Around Copyright Protection of Reverse Engineering in the Software Industry*, 6 CHI.-KENT J. INTELL. PROP. 66 (2005).

115. *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317 (Fed. Cir. 2003).

116. *Id.*

**V. BOWERS v. BAYSTATE TECHNOLOGIES, INC.**

Harold L. Bowers (Bowers) created a template to improve command selection within computer aided design (CAD) software.<sup>117</sup> Bowers received a patent for his template in 1990, and commercialized the template as “Cadjet” for use with “CADKEY.”<sup>118</sup> Bowers received a reexamination certificate in 1997 because of prior art concerns.<sup>119</sup>

An engineer by the name of George W. Ford III (Ford), envisioned a way to improve the template and CAD software.<sup>120</sup> Ford designed “Geodraft” to allow engineers to insert technical tolerances into CAD designs.<sup>121</sup> Geodraft allowed a design to comply with American National Standards Institute (ANSI) tolerance requirements.<sup>122</sup> Ford registered and obtained a copyright covering Geodraft.<sup>123</sup> In 1989, Ford offered Bowers an exclusive license to his Geodraft software, which allowed Bowers to bundle Geodraft and Cadjet together as the “Designer’s Toolkit.”<sup>124</sup> Bowers included a shrink-wrap license prohibiting reverse engineering of the Designer’s Toolkit.<sup>125</sup> Around the same time, Baystate created and marketed tools for CADKEY.<sup>126</sup> One of those tools, known as “Draft-Pak” version 1 and 2, also featured template and tolerance software similar to the Designer’s Toolkit. Aware of Draft-Pak, Bowers attempted to offer his template to Baystate to bundle with Draft-Pak in 1988 and 1989, but Baystate declined saying that it had “the in-house capability to develop the type of products . . .” that Bowers had proposed.<sup>127</sup>

In 1990, Bowers released Designer’s Toolkit, and by 1991, Baystate had obtained copies. Shortly after obtaining the copies of Bowers’ product, Baystate released Draft-Pak version 3, which incorporated several of the features of Designer’s Toolkit.<sup>128</sup> Understandably, the introduction of Baystate’s Draft-Pak version 3 prompted an intense price competition between Bowers and Baystate. Bowers attempted to propel himself past Baystate in the market by negotiating with Cadkey, Inc. to provide the Designer’s Toolkit with CADKEY for free.<sup>129</sup> Bowers also planned to sell software upgrades to CADKEY customers in hope that he

---

117. *Id.* at 1320.

118. *Id.* at 1321.

119. *Id.*

120. *Id.*

121. *Id.* at 1322.

122. *Id.*

123. *Id.*

124. *Id.*

125. *Id.*

126. *Id.*

127. *Id.*

128. *Id.*

129. *Id.*

could recoup his profits.<sup>130</sup> Baystate pressured Cadkey, Inc. to retract their agreement with Bowers and eventually Baystate purchased Cadkey, Inc., completely “eliminat[ing] Mr. Bowers from the CADKEY network—effectively preventing him from developing and marketing the Designer’s Toolkit for that program.”<sup>131</sup>

In 1991, Baystate sued Bowers for declaratory judgment on the grounds that Baystate’s products do not infringe Bowers’ patent because it was invalid and unenforceable.<sup>132</sup> Following trial, the jury found for Bowers on all claims, and the district court set aside copyright damages as duplicative of contract damages.<sup>133</sup> Baystate subsequently filed motions for judgment as a matter of law or for a new trial.<sup>134</sup> The district court denied Baystate’s motions for judgment as a matter of law or for a new trial.<sup>135</sup> Consequently, Baystate appealed the district court’s denial of its motions.<sup>136</sup> Bowers appealed the district court’s denial of copyright damages.<sup>137</sup>

On appeal, Baystate argued that the Copyright Act preempted Bowers’ shrink-wrap license agreement that prohibited reverse engineering.<sup>138</sup> The court held that the Copyright Act does not preempt or narrow the scope of Bowers’ contract claim.<sup>139</sup>

Judge Rader’s majority opinion looked to First Circuit case law and found no real stance on whether copyright law should preempt a state law contract.<sup>140</sup> He focused his rationale on the freedom to contract. He made note of the First Circuit case, *Data Gen. Corp. v. Grumman Sys. Support Corp.*, where Data General alleged that Grumman misappropriated its trade secret software by obtaining the software from customers and former employees who were bound by confidentiality agreements.<sup>141</sup> Grumman argued that the Copyright Act preempted Data General’s trade secret claim.<sup>142</sup> The First Circuit held that the Copyright Act did not preempt the trade secret claim because the elements of the state law required elements that pushed the claim outside the scope copyright preemption.<sup>143</sup> So long as a state law claim required additional elements

---

130. *Id.*

131. *Id.*

132. *Id.*

133. *Id.*

134. *Id.*

135. *Id.*

136. *Id.*

137. *Id.*

138. *Id.* at 1323

139. *Id.*

140. *Id.* at 1324.

141. *Id.*

142. *Id.*

143. *Id.*

of proof beyond mere copying, the state law claim should escape preemption.<sup>144</sup>

Judge Rader followed the reasoning of *Data General* and adapted it to the shrink-wrap agreement used by Bowers.<sup>145</sup> He highlighted the reliance on contract elements in the reasoning of *ProCD, Inc. v. Zeidenberg* and agreed that mutual assent and consideration make contract claims considerably different than copyright claims.<sup>146</sup> In *ProCD*, ProCD compiled more than 3,000 telephone directories into a database called SelectPhone and sold it on CD-ROM discs.<sup>147</sup> All boxes containing the CD had a shrink-wrap license agreement outlining restrictions of use. Zeidenberg purchased the CD and made the database available to anyone online willing to pay for it.<sup>148</sup> The appellate court held that a shrink-wrap license agreement included with software is a legally binding contract under the Uniform Commercial Code (UCC).<sup>149</sup> The court reasoned: “A copyright is a right against the world. Contracts, by contrast, generally affect only their parties; strangers may do as they please, so contracts do not create ‘exclusive rights.’”<sup>150</sup>

Judge Rader dismissed the Fifth Circuit court in *Vault Corp. v. Quaid Software, Ltd.*, ruling that a state law prohibiting the copying of a computer program is preempted by the Copyright Act.<sup>151</sup> Instead, he believed that private contractual agreements, including shrink-wrap licenses, were supported by mutual assent and consideration pushing them outside the scope of the ruling in *Vault*.<sup>152</sup> Judge Rader observed that most courts have found that the Copyright Act does not preempt contractual restrictions on copyrighted works, and therefore held that the Copyright Act did not preempt Bowers’ contract claims.<sup>153</sup>

It is important to note that Judge Rader never claimed that reverse engineering was not defensible by fair use. In fact, he agreed with the ruling in *Atari* and held that reverse engineering indeed constituted fair use.<sup>154</sup> Instead, the focus was on the discrepancy between contract claims and copyright claims.<sup>155</sup>

After finding that the Copyright did not preempt Bowers’ contract claims, Judge Rader focused his opinion on whether Baystate breached

---

144. *Id.*

145. *Id.*

146. *Id.* (citing *ProCD, Inc. v. Zeidenberg*, 86 F.3d 1447, 1454 (7th Cir. 1996)).

147. *Zeidenberg*, 86 F.3d at 1449.

148. *Id.*

149. *Id.*

150. *Bowers*, 320 F.3d at 1325 (citing *Zeidenberg*, 86 F.3d at 1454).

151. *Id.*

152. *Id.*

153. *Id.*

154. *Id.* at 1325.

155. *Id.*

the license agreement.<sup>156</sup> The shrink-wrap agreement unambiguously prohibited reverse engineering, and Baystate agreed willingly and freely once they opened the software.<sup>157</sup> Baystate's CEO, Robert Bean, admitted that Baystate frequently analyzed competitors' products to duplicate their technology, and the court found extensive evidence showing the similarities between Geodraft and Draft-Pak—all alluding to the fact that Baystate did indeed reverse engineer Bowers' software, thus breaching the license agreement.<sup>158</sup>

Judge Dyk's dissenting opinion, with regards to fair use, begins with the purpose of the Copyright Act, which is “to promote the Progress of Science and useful Arts . . . ,” and the importance of limiting copyright protection through a fair use defense.<sup>159</sup> He sees the fair use defense as a facilitator to those individuals who reverse engineered products ultimately to progress and innovate useful technology.<sup>160</sup> Judge Dyk believes that an author should not be afforded protection for an idea just because the idea is embodied in a computer program.<sup>161</sup> He emphasizes the policy that copyright protection does not extend to ideas, procedures, processes, systems, and methods of operations because he believes that is the function of patent protection.<sup>162</sup>

## VI. THE INADEQUATE SOLUTION: FAIR USE

The language of Section 102(a) of the Copyright Act specifically states that “[c]opyright 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. . . .”<sup>163</sup> The language of Section 102 of the Copyright Act points to Congress's intent to establish copyright law as a progressive law that would encompass protection for those works of authorship listed in Section 102.<sup>164</sup> *Bowers* suggests that there is balance lacking in the realm of copyright law. The realistic situation is that the protection offered to computer software is ambiguous. In *Bowers*,

---

156. *Id.* at 1326.

157. *Id.*

158. *Id.* at 1327.

159. *Id.* at 1336.

160. *Id.*

161. *Id.* (citing *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 842 (Fed. Cir. 1992)).

162. *Id.*

163. 17 U.S.C. § 102(a) (1976) (emphasis added).

164. *Id.* (stating that “copyright protection subsists in any tangible medium of expression . . . *now known or later developed*” (emphasis added)).

the Copyright Act did not serve as a shield, but rather as a sword for Baystate. Both Judge Rader and Judge Dyk agreed that Baystate's reverse engineering of Bowers' program was fair use. The only element that pushed Bowers to ultimately win the case was Bowers' contract. If Bowers had never contracted with Baystate to prohibit Baystate from reverse engineering Bowers' program, *Bowers* would have likely had a different outcome.

Copyright law prevents others from using copyrighted works without authorization, yet it allows copied works to be profited from. In *Bowers*, two companies were trying to progress innovation. Reverse engineering allowed Baystate to disassemble Bowers' code and create a product that ultimately pushed Bowers out of the market.<sup>165</sup> The goal of copyright law is "[t]o promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries;" while providing financial benefit to copyright owners.<sup>166</sup> It is clear that Bowers lost the financial benefit that he was supposed to be afforded when Baystate reverse engineered Bowers' program.<sup>167</sup> Finding a balance between respecting an author's desire to protect its intellectual property and promoting innovation through reverse engineering is always difficult. Judge Rader's opinion in *Bowers* seemed to protect Bowers' software by raising contract principles above fair use. However, there had already been many cases at the federal level establishing that reverse engineering could amount to fair use before *Bowers* was decided.<sup>168</sup>

The longstanding Copyright Act of 1976, and its 1980 amendment that defined a "computer program," is around 35 years old. Since the 1980 amendment, the computer software industry has seen an exponential growth, gaining billions of dollars.<sup>169</sup> Anthony L. Capes recognized the importance of the computer software industry back in 1993:

---

165. *Bowers*, 320 F.3d at 1322.

166. U.S. CONST. art. I, § 8, cl. 8. *See* 17 U.S.C. § 106(3) (1976) (stating that copyright owners have the exclusive right to distribute copies of the copyrighted work by sale).

167. *Bowers*, 320 F.3d at 1322 (explaining that Baystate's purchase of CadKey, Inc., eliminated Bowers from the CADKEY network and prevented him from develop and marketing his Designer's Toolkit for that program).

168. *See, e.g., Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 843 (Fed. Cir. 1992) ("When the nature of a work requires intermediate copying to understand the ideas and processes in a copyrighted work, that nature supports a fair use for intermediate copying. Thus, reverse engineering object code to discern the unprotectable ideas in a computer program is a fair use"); *Sega Enters., Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992); *Sony Computer Entm't, Inc. v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000); *Bateman v. Mnemonics, Inc.*, 79 F.3d 1532 (11th Cir. 1996).

169. *See generally Software Industry Facts and Figures*, Bus. Software Alliance, [http://www.bsa.org/country/Public%20Policy/~media/Files/Policy/Security/General/sw\\_factsfigures.ashx](http://www.bsa.org/country/Public%20Policy/~media/Files/Policy/Security/General/sw_factsfigures.ashx) (last visited May 5, 2015) (showing various charts and graphs describing the growth of the computer software industry).

IBM's new notebook computers. . . . [have] the processing, memory capacity, and disk capacity equivalent of a computer that would have taken up [an entire room] years ago. [It] costs at most a few thousand dollars where that earlier computer probably cost a few million dollars. [W]hat that march of technology means is that, more so than ever before, the computer business is driven by software, not by hardware. [Personal and notebook computers] are being sold in the millions of units. The big main frame computers sold in, perhaps, the tens of thousands.<sup>170</sup>

The significance of protecting software has since become even more apparent considering that an individual can make millions, if not billions, of dollars on cell phone application software.<sup>171</sup> In an industry where vast amounts of money are at stake, copyright law provides computer software copyright owners only limited protection with regards to its source code.

Courts have always struggled with trying to define a line between unprotectable ideas and protectable expression.<sup>172</sup> In fact, Judge Learned Hand stated in *Nichols v. Universal Pictures Corp.* that

[n]obody has ever been able to fix that boundary, and nobody ever can. In some cases the question has been treated as though it were analogous to lifting a portion out of the copyrighted work; but the analogy is not a good one, because, though the skeleton is a part of the body, it pervades and supports the whole.<sup>173</sup>

The issue becomes even more complex when dealing with computer programs because the way that a program is expressed, and the function that it provides, are almost equivalent considering that a program's function is defined by how the program's code is written. Therefore, when a court states that "reverse engineering object code to discern the unprotectable ideas in a computer program is a fair use," that reasoning fails to recognize that the program's object code itself is an original compilation that creates a workable program that is protected by copyright law.<sup>174</sup> It is well established that a program, whether in object

---

170. Symposium, *Copyright Protection: Has Look & Feel Crashed?*, 11 CARDOZO ARTS & ENT. L.J. 721, 743 (1993).

171. See Drake Baer & Skye Gould, *Here are the 20 Fastest Companies to Reach a \$2 Billion Valuation* (May 1, 2015, 10:38 AM), <http://www.businessinsider.com/fastest-companies-to-reach-a-2-billion-valuation-2015-5> (showing a graphic chart of 20 startup companies that reached a \$2 billion valuation, many of which developed cell phone applications).

172. Christopher Hager, *Apples & Oranges: Reverse Engineering as a Fair Use After Atari v. Nintendo and Sega v. Accolade*, 20 RUTGERS COMPUTER & TECH. L.J. 259, 318 (1994) (citing *Sega*, 977 F.2d at 1524).

173. *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d. Cir. 1930).

174. Hager, *supra* note 172, at 319 (citing *Atari*, 975 F.2d at 843; see also *Apple Computers*,

code or source code, is protected from unauthorized copying.<sup>175</sup>

An individual that reverse engineers a copyright owner's program should not be allowed to use copyright's fair use exception as a sword to allow that individual to use the reverse engineered source code to create a program that ultimately disrupts the copyright owner's right to disseminate its work to the public. Allowing an individual to reverse engineer a program's object code to discern its source code can be seen as "promot[ing] the Progress of Science and useful Arts," but it also creates competition that undermines the copyright owner's right to profit off of their work and contradicts the directive to "secur[e] for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries."<sup>176</sup>

The U.S. Supreme Court has stated that "[t]he economic philosophy behind the clause empowering Congress to grant patents and copyrights is the conviction that encouragement of individual effort by personal gain is the best way to advance public welfare through the talents of authors and inventors in 'Science and useful Arts.'"<sup>177</sup> Software developers seek to create new and innovative programs because they believe that their programs can provide an improvement that other developers have yet to discover. Creating new and innovative programs gives a developer a competitive advantage in the marketplace. By allowing the fair use defense for reverse engineering, a competitor can reverse engineer a program to create a competitive work, negating any advantage the original copyright owner had. Accordingly, Judge Becker wrote in his majority opinion in *Whelan* that, "one can approximate a program and thereby gain a significant advantage over competitors even though additional work is needed to complete the program."<sup>178</sup> Losing the advantage in the market place can stifle a developer's incentive to create new and innovative programs. Without an incentive to create, the advancement of the public welfare will also be stifled.

Having a competitive advantage does not mean that the copyright owner has a monopoly. An element of owning a monopoly is having the power to prevent competition.<sup>179</sup> The idea that the reverse engineering fair use defense helps to negate the monopolistic elements that computer software could be afforded through patent-like protection is quite

---

Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1249 (3d Cir. 1983) (holding computer object code protectable as part of the overall copyrightable program)).

175. See *Apple*, 714 F.2d at 1249; *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1233 (3d Cir. 1986).

176. U.S. CONST. art. I, § 8, cl. 8. See also 17 U.S.C. § 106(3) (stating that copyright owners have the exclusive right to distribute copies of the copyrighted work by sale).

177. *Mazer v. Stein*, 74 U.S. 460, 471 (1954).

178. *Whelan*, 797 F.2d at 1237.

179. Symposium, *supra* note 170, at 748.

illegitimate. In fact, a major premise of patent law is that the invention being claimed must be publicly disclosed.<sup>180</sup> If an individual sought to understand how a patented invention was made, that individual could simply look up the patent and view its specification.<sup>181</sup> Whereas in copyright law, there is no initiative that proclaims that copyright holders, and more specifically computer software developers, must disclose their invention to the public. In the *Bowers* dissenting opinion, Judge Dyk believes that the fair use defense is necessary so that copyright protection does not extend patent-like protection to the copyright holder.<sup>182</sup> Judge Dyk fails to see that eliminating the fair use defense for reverse engineering does not create patent-like protection for a computer program, nor does it stop an individual from creating similar works. Instead, it allows a copyright owner to prohibit individuals from reverse engineering their programs so that an individual could not copy, distribute, or create a derivative program based off of the copyrighted program without the owners' consent. It seems as though Judge Dyk is using the fair use defense as a way to justify the equitability of copyright law protections as compared to patent law protections for computer software. Yet, Judge Dyk intends on distinguishing copyright law from patent law.

Computer software is inherently floating in legal limbo due to its functional nature. Some individuals are reluctant to give computer software a home in the realm of patent law due to the possibility of stifling innovation, while others feel that copyright law provides a shelter that is too inadequate to properly house software's unique characteristics.<sup>183</sup> In the recent appellate decision, *Oracle Am., Inc. v. Google, Inc.*, Google propelled an argument that copyright law is no longer suitable to protect software programs, leaving patent law as a more applicable source of protection for computer programs.<sup>184</sup> The court rejected Google's

---

180. 35 U.S.C. § 112(a) (2012).

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same, and shall set forth the best mode contemplated by the inventor or joint inventor of carrying out the invention.

*Id.*

181. *Id.*

182. *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317, 1336 (Fed. Cir. 2003).

183. See generally Symposium, *supra* note 170, at 721.

184. *Oracle Am., Inc. v. Google, Inc.*, 750 F.3d 1339, 1380 (Fed. Cir. 2014) (stating that “[a]fter Sega, developers could no longer hope to protect [software] interfaces by copyright . . . Sega signaled that the only reliable means for protecting the functional requirements for achieving interoperability was by patenting them.” (quoting Pamela Samuelson, *Are Patents on Interfaces*

argument citing various authorities suggesting that computer software should be protected by copyright law.<sup>185</sup> Additionally, the court believed that the interoperability of replicated software has no relevance to answering the question of whether programs are copyrightable.<sup>186</sup> Instead, the court suggested that an interoperability analysis is better suited for a fair use inquiry.<sup>187</sup> The court relied on the idea that the motive to achieve compatibility does not implicate the issue of whether ideas and expressions merge, but instead the process of achieving compatibility has commercial and competitive roots, which are more applicable to the first factor in the fair use analysis requiring a consideration of the commercial nature of the use.<sup>188</sup> The court suggested, and the former Register of Copyrights of the United States agreed in his brief amicus curiae that, had this been a case where the programs in dispute were reverse engineered to discern their ideas and functionality, and then created a new structure and code were subsequently created, then there would be no remedy under copyright law.<sup>189</sup> The court concluded with a resonating statement: “Until either the Supreme Court or Congress tells us otherwise, we are bound to . . . afford software programs protection under the copyright

---

*Impeding Interoperability?*, 93 MINN. L. REV. 1943, 1959 (2009))).

185. See *Oracle*, 750 F.3d at 1370 (citing *Technology Quarterly, Stalking Trolls*, ECONOMIST, Mar. 8, 2014, <http://www.economist.com/news/technology-quarterly/21598321-intellectual-property-after-being-blamed-stymying-innovation-america-vague>).

[M]any innovators have argued that the electronics and software industries would flourish if companies trying to bring new technology (software innovations included) to market did not have to worry about being sued for infringing thousands of absurd patents at every turn. A perfectly adequate means of protecting and rewarding software developers for their ingenuity has existed for over 300 years. It is called copyright.”)

*Id.*; Timothy B. Lee, *Will the Supreme Court Save Us from Software Patents?*, WASH. POST (Feb. 26, 2014, 1:13 PM), <http://www.washingtonpost.com/blogs/the-switch/wp/2014/02/26/will-the-supreme-court-save-us-from-software-patents/>.

If you write a book or a song, you can get copyright protection for it. If you invent a new pill or a better mousetrap, you can get a patent on it. But for the last two decades, software has had the distinction of being potentially eligible for both copyright and patent protection. Critics say that’s a mistake. They argue that the complex and expensive patent system is a terrible fit for the fast-moving software industry. And they argue that patent protection is unnecessary because software innovators already have copyright protection available.

*Id.*

186. Lee, *supra* note 185.

187. *Id.* at 1371.

188. *Id.*

189. *Id.* at 1370.

laws.”<sup>190</sup>

Interestingly, the U.S. Supreme Court has denied certiorari to both *Bowers* and *Oracle*, allowing the courts to continue to manifest their own solutions to the reverse engineering issue and the extent of copyright protection for computer programs. The lack of interest by the U.S. Supreme Court is vexing, considering the absence of clarity that persists with regards to computer software copyright protection. Perhaps the U.S. Supreme Court has a legitimate reason for not delving into the challenging task of clarifying the appropriate extent of the fair use defense, possibly due to an apprehensiveness concerning the repercussions that could result from incorrectly towing the thin line between liberal and conservative copyright protection. As of now, courts should not focus on whether copyright law provides adequate protection for computer software, because that decision must be tackled by either Congress or the U.S. Supreme Court. Until there is explicit legislation and precedent that governs this issue, the present and future courses of action should focus on the assimilation of computer software into copyright law, where courts convey rulings that allow for the legitimate protection of the copyright holder’s rights. Accordingly, granting copyright owners the power to make copies or to authorize copying, distributing, or creating a derivative work based on the copyrighted work are all basic tenants of copyright law, and as it stands now, allowing the fair use defense for reverse engineering disrupts those tenants.<sup>191</sup> Therefore, courts should be reluctant to apply the fair use defense for reverse engineering due to the danger of protecting computer software too narrowly, which ultimately could impede the incentive for developers to create new programs, and subsequently, thwart the progression of innovation.

## VII. CONCLUSION

Computer software developers are wondering if they will ever receive the protection that their programs deserve. Congress does not respect the fact that computer programs are functional by nature, and instead considers programs to be an “expression” because they are created with a “language,” thus positioning programs under the umbrella of copyright law.<sup>192</sup> In *Sega*, Judge Reinhart confirmed that computer programs are not so simply “literary works.” “[C]omputer programs are, in essence,

---

190. *Id.* at 1380.

191. 17 U.S.C. § 106 (1976) (granting copyright owners the exclusive right to copy, distribute, or prepare derivative works of the copyrighted work).

192. *See, e.g.,* Apple Computers, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1249 (3d Cir. 1983); Dogan & Liu, *supra* note 32, at 208.

utilitarian articles—articles that accomplish tasks. As such, they contain many logical, structural, and visual display elements that are dictated by the function to be performed. . . .”<sup>193</sup> Copyright law does not afford computer software sufficient protection, so little in fact that it allows others to copy a copyright owner’s program and reverse engineer it to create a work that could cause a loss in financial benefit to the copyright holder. In *Altai*, Judge Walker stated that he believed copyright law might not be appropriate for a creation such as computer software: “Generally, we think that copyright registration—with its indiscriminating availability—is not ideally suited to deal with the highly dynamic technology of computer science. Thus far, many of the decisions in this area reflect the courts’ attempt to fit the proverbial square peg in a round hole.”<sup>194</sup> Authors of computer programs are provided with the inadequate protection of copyright law that is supplemented by further limiting a copyright owner’s rights by allowing reverse engineering to constitute a fair use defense.

The fair use doctrine was set in place to provide a sense of balance between the exclusive rights granted to the author while promoting the progress of science and the useful arts.<sup>195</sup> But, as we have seen in *Bowers*, the results of allowing fair use can invoke consequences that inevitably injure the author whose goal is to provide an innovative product to the public domain.<sup>196</sup> Courts should be careful to narrow the scope of copyright protection for computer programs because it could result in a lack of incentive for the author, and ultimately, a lack of progress and innovation within society. Maybe, as Judge Walker says, this issue will be resolved from further legislative investigation—possibly a CONTU II.<sup>197</sup> Until then, courts should steer away from using the fair use doctrine to defend reverse engineering.

---

193. *Sega Enters., Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th Cir. 1992).

194. *Computer Assocs. Int’l, Inc. v. Altai, Inc.*, 982 F.2d 693 (2d Cir. 1992).

195. *Bowers v. Baystate Techs., Inc.*, 320 F.3d 1317, 1335 (Fed. Cir. 2003) (citing U.S. CONST. art I, § 8, cl. 8).

196. *Id.* at 1322 (explaining that Baystate’s purchase of CadKey, Inc., eliminated Bowers from the CADKEY network and prevented him from develop and marketing his Designer’s Toolkit for that program).

197. *Altai*, 982 F.2d at 712.