

Indiana Law Review

Volume 52

2019

Number 2

NOTES

PROTECTING SOFTWARE INTELLECTUAL PROPERTY AFTER THE *LEXMARK* DECISION

NICOLE L. DOBIAS*

INTRODUCTION

Software surrounds us; it's integrated into almost every aspect of life.¹ Each morning, we wake up to the alarm clock function on our smartphone, Amazon Echo, or Google Home. We scroll through news and social media applications on our iPhones as we roll out of bed. When driving to work, we listen to music or podcasts downloaded onto our iPhones through the car's Bluetooth connection. Upon arriving at work, we check our Google calendars, which contain the meetings and deadlines that dictate our days. After work, we hit the gym, running on treadmills that contain personal TVs attached to the machines. We then head home and order food for dinner, using a food delivery application on our iPhones. We end our days binge-watching our favorite TV shows on Netflix or ordering an Uber to go meet up with friends.

Software provides a large part of the modern experience, running on more devices than traditional desktop and laptop computers.² In fact, software runs “everything from coffeemakers and telephones to car brakes, airplanes, and sophisticated medical imaging equipment.”³ In addition to the effect software has on day-to-day life, it also has an impact on the economy.⁴

Specific law and policy regarding copyright and patent protection surround software.⁵ Because products of information and technology, such as software, are

* Nicole Dobias, JD Candidate, 2019, Indiana University Robert H. McKinney School of Law; B.A. 2014, DePauw University – Greencastle, Indiana. I would like to thank Professor Lea Shaver for her invaluable insight, wisdom, and guidance throughout the Note writing process, as well as editors of the *Indiana Law Review* for their hard work in helping me develop this Note. I would also like to thank my family and friends for their support and encouragement.

1. See MICHAEL L. RUSTAD, SOFTWARE LICENSING, CLOUD COMPUTING AGREEMENTS, OPEN SOURCE, AND INTERNET TERMS OF USE § 1.01 (2016).

2. XUAN-THAO N. NGUYEN ET AL., INTELLECTUAL PROPERTY, SOFTWARE, AND INFORMATION LICENSING: LAW AND PRACTICE 510 (2006).

3. *Id.*

4. See Karsten Strauss, *Why Tech Grows in Indianapolis*, FORBES (July 27, 2016, 04:12 PM), <https://www.forbes.com/sites/karstenstrauss/2016/07/27/why-tech-grows-in-indianapolis/> [<https://perma.cc/43SU-ADC3>].

5. See ROBERT W. GOMULKIEWICZ ET AL., LICENSING INTELLECTUAL PROPERTY: LAW AND APPLICATION (2008).

intangible, licensing is the predominant transactional tool.⁶ Licensing law generally consists of contract law, but intellectual property law plays a predominant part in how licensing law governs and functions.⁷ Specifically, a license is essentially a contract about copyright and patent law.⁸ Although copyright and patent law are codified, both the extent of patent law protection afforded to software⁹ and exhaustion of rights associated with patent and copyright law have been defined by the common law.¹⁰

In 2014, the Supreme Court increased the difficulty of patenting computer-implemented inventions, such as software, with its holding in *Alice Corp. v. CLS Bank International*.¹¹ The Court held that implementing an abstract idea, such as a method or system, on a computer does not sufficiently transform the abstract idea into patent-eligible subject matter.¹² Since the case's publication, "*Alice* has proven 'deadly' for software patents."¹³

On May 30, 2017, the Supreme Court further disincentivized protecting software through patents when it clarified the patent exhaustion doctrine, holding in *Impression Products, Inc. v. Lexmark International, Inc.* that when a patent owner sells his or her patented product, the owner relinquishes all rights associated with that product—including any restrictions the patentee wishes to impose on the purchaser.¹⁴ The Supreme Court's reasoning in this case carries a variety of impacts. The *Lexmark* decision will impact the software industry in the way our devices are produced and sold to us—devices we depend upon every day, such as cell phones.¹⁵ Additionally, this outcome will influence the way software companies choose to do business and expand.¹⁶ Finally, when the Supreme Court handed down this decision, it furthered the decline of protecting software through patents started by *Alice*.¹⁷

Alice caused a dramatic decrease in patenting of software by raising the patent-eligibility standard for software and other computer-implemented

6. *Id.* at 3.

7. *Id.* at 8.

8. *Id.* at 13.

9. *Alice Corp. v. CLS Bank Int'l*, 573 U.S. 208, 226 (2014).

10. Vincent Chiappetta, *Patent Exhaustion: What's it Good For?*, 51 SANTA CLARA L. REV. 1087, 1088 (2011).

11. *See Alice*, 573 U.S. 208.

12. *Id.* at 217-21.

13. Jasper L. Tran, *Two Years after Alice v. CLS Bank*, 98 J. PAT. & TRADEMARK OFF. SOC'Y 354 (2016) [hereinafter *Two Years after Alice*] (quoting Foley Hoag LLP, *The Post Office Gets into the Alice Act*, PTAB BLOG (Oct. 27, 2015), <http://www.ptab-blog.com/2015/10/27/the-post-office-gets-into-the-alice-act/> [https://perma.cc/QC7Q-UTMJ]), *cf.* Jasper L. Tran, *Software Patents: A One-Year Review of Alice v. CLS Bank*, 97 J. PAT. & TRADEMARK OFF. SOC'Y 532, 533 (2015) [hereinafter *One Year after Alice*].

14. *Impression Prods. v. Lexmark Int'l, Inc.*, 137 S. Ct. 1523, 1529 (2017).

15. NGUYEN ET AL., *supra* note 2, at 532.

16. *See Strauss*, *supra* note 4.

17. *See Alice*, 573 U.S. 208.

inventions.¹⁸ *Lexmark* will cause a further decrease in patenting of software because software companies may decide that patenting their computer-implemented inventions is not worth pursuing if protection ends upon the first sale of the product.¹⁹ As a result of such boundaries imposed by the courts, software companies must either find a way to utilize copyright law to protect their products, or push for the creation of a *sui generis* system of intellectual property law to govern software.

This Note argues that a different approach is needed to fill the void left by the *Alice* and *Lexmark* decisions. Part I outlines how copyright and patent law interact to protect software. Part II introduces the two Supreme Court decisions: *Alice* and *Lexmark*. Part III explains the impact of *Alice* on software patenting since 2014. Part IV argues that the *Lexmark* decision will have a similar negative impact. In light of the Supreme Court's trend to remove patent protection from the software industry, Part V argues that a different form of intellectual property protection is needed. This Part highlights the possibility of using copyright law to achieve robust intellectual property protection but notes the flaws with this approach. As a result, Part V argues that the time has come to create a special regime of intellectual property law to specifically govern software.

I. INTELLECTUAL PROPERTY AND THE SOFTWARE INDUSTRY

A. Software

Section 101 of Title 17 of the United States Code defines a computer program as “a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.”²⁰ Computer programs fall into two categories regarding functionality: Application programs and operating programs.²¹ Application programs perform certain tasks for the user, including “word processing, checkbook balancing, or playing a game.”²² Examples of application programs include Microsoft Word and Microsoft Excel. Operating programs either facilitate the use of application programs or manage internal computer functions.²³ Examples of operating systems include Apple's macOS, Google's Android OS, and Linux operating system. Both copyright²⁴ and patent law²⁵ provide protection for software programs.

18. See *Two Years after Alice*, *supra* note 13.

19. *Lexmark*, 137 S. Ct. 1523.

20. 17 U.S.C. § 101 (2018).

21. *Apple Comput., Inc. v. Franklin Comput. Corp.*, 714 F.2d 1240, 1243 (3d Cir. 1983).

22. *Id.*

23. *Id.*

24. See *Apple Comput., Inc. v. Microsoft Corp.*, 35 F.3d 1435 (9th Cir. 1994); *Franklin*, 714 F.2d 1240.

25. NGUYEN ET AL., *supra* note 2, at 507.

B. Copyright Law

Copyright protection exists “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.”²⁶ Such protection extends to a variety of works.²⁷ The Copyright Act lays out a list of seven categories of works or authorship in which copyrighted works fall into: literary works; pictorial, graphic and sculptural works; architectural works; dramatic, pantomime and choreographic works; musical works and sound recordings; motion pictures and other audiovisual works; derivative works and compilations.²⁸

Copyright protection extends to computer programs as literary works.²⁹ Literary works include “works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied.”³⁰ Although the Copyright Act does not explicitly include computer programs as works of authorship, the legislative history indicates these programs should be considered copyrightable under literary works.³¹ Consequently, copyright law protects software and computer programs.³²

Importantly, the Copyright Act excludes functional or useful articles from copyright protection, including systems, methods, processes and procedures.³³ This may seem problematic because both application programs and operating programs contain a series of integrated methods of instructions, which would ordinarily be excluded from copyright protection.³⁴ However, courts have continuously upheld the copyrightability of the instructions contained in both types of programs as protectable literary works.³⁵

Copyright law protects both the literal and non-literal elements of computer programs.³⁶ Literal elements of computer programs include the source and object code.³⁷ Source code refers to instructions or “spelled-out program commands that

26. 17 U.S.C. § 102(a) (2018).

27. *Id.*

28. *Id.*

29. *Franklin*, 714 F.2d at 1249.

30. 17 U.S.C. § 101.

31. *Franklin*, 714 F.2d at 1247.

32. *Apple Comput., Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1447 (9th Cir. 1994).

33. 17 U.S.C. §§ 101-102. Section 102 excludes from copyright protection “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.”

34. *See Franklin*, 714 F.2d at 1251.

35. *Id.*; *see also* DAVID NIMMER, 1 NIMMER ON COPYRIGHT § 2A.10 (2017).

36. *Comput. Assocs. Int'l v. Altai*, 982 F.2d 693, 702 (2d Cir. 1992).

37. *Id.*; *see generally* NIMMER, *supra* note 35, § 2A.10.

humans can read,”³⁸ and object code refers to binary language “comprised of zeros and ones through which the computer directly receives its instructions.”³⁹ Non-literal elements of the computer program include the “program’s sequence, structure, and organization, as well as the program’s user interface.”⁴⁰ Although courts consistently afford copyright protection to the literal elements, the extent of protection afforded to the non-literal elements “depends on whether, on the particular facts of each case, the component in question qualifies as an expression of an idea, or an idea itself;”⁴¹ copyright protection extends only to expressions of ideas, not the ideas themselves.⁴²

Copyright owners enjoy a variety of rights regarding their copyrighted works.⁴³ Specifically, copyright owners retain the rights to make copies of the work, prepare derivative works, distribute copies of the work, publicly perform the work, and display the copyrighted work publicly.⁴⁴ However, these rights are generally exhausted upon the first sale of a copy of the copyrighted work—a concept known as the first sale doctrine.⁴⁵ The first sale doctrine encompasses the idea that “once the copyright owner places a copyrighted item in the stream of commerce by selling it, he has exhausted his exclusive statutory right to control its distribution.”⁴⁶ As such, an owner of a copy of a copyrighted work can “sell or otherwise dispose of” the copy.⁴⁷ In fact, upon the first sale of a copyrighted item, any following purchaser is considered an owner of that item, and that owner can sell the item without permission from the copyright owner.⁴⁸

C. Patent Law

Section 101 of the Patent Act provides patent protection for any “useful process, machine, manufacture, composition of matter, or any new and useful improvement thereof.”⁴⁹ However, courts exclude abstract ideas, laws of nature, and natural phenomena from patent protection.⁵⁰ As software developed over the past half-century, courts struggled to establish the contours and boundaries of its patent protection.⁵¹ All software—regardless of type or programming

38. *Lexmark Int’l, Inc. v. Static Control Components, Inc.*, 387 F.3d 522, 533 (6th Cir. 2004).

39. *Altai*, 982 F.2d at 698.

40. *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1355-1356 (Fed. Cir. 2014).

41. *Johnson Controls, Inc. v. Phoenix Control Sys.*, 886 F.2d 1173, 1175 (9th Cir. 1989).

42. 17 U.S.C. § 102 (2018); *Feist Publ’ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 350 (1991).

43. 17 U.S.C. § 106.

44. *Id.*

45. *Id.* § 109(a).

46. *Quality King Distribs. v. L’Anza Research Int’l*, 523 U.S. 135, 152 (1998).

47. 17 U.S.C. § 109(a).

48. *Quality King*, 523 U.S. at 145.

49. 35 U.S.C. § 101 (2018).

50. *Ass’n for Molecular Pathology v. Myriad Genetics, Inc.*, 569 U.S. 576, 589 (2013).

51. *Ognjen Zivojnovic, Patentable Subject Matter after Alice—Distinguishing Narrow*

language—consists of a combination of algorithms, containing “a series of logical or arithmetic steps performed on loaded and stored numbers.”⁵² But courts historically consider such algorithms to be abstract ideas, and patent-ineligible as a result.⁵³ Nevertheless, some applications of abstract ideas may be entitled to patent protection.⁵⁴ The concept of patent protection of software is further addressed in Parts II and III of this Note.⁵⁵

Patent law gives patentees the “right to exclude others from making, using, offering for sale, or selling [their] invention[s].”⁵⁶ Patent law seeks to promote scientific advancement by granting a limited monopoly to inventors, which also allows inventors to enjoy the financial benefits and rewards for their work.⁵⁷ However, once an inventor sells his patented product, he has enjoyed such benefits and rewards for his work by the limited monopoly.⁵⁸ The patent owner’s sale of his patented product meets the purpose of patent law, and there remains no basis for constraining others’ use and appreciation of the patented product.⁵⁹

As a result, the first-sale doctrine—also known as patent exhaustion—dictates that once a customer purchases a patented product, the customer may use or resell the product as he or she pleases.⁶⁰ A sale of the patented product by the patentee “exhausts” the patentee’s rights regarding further control of the patented product.⁶¹ Further, these sales transfer to the purchasers the rights to use, dispose of, and resell the products and protect the purchasers from an infringement suit.⁶² The patent exhaustion doctrine requires an authorized sale to be unconditional and unrestricted, meaning “without any express conditions on the sale or license.”⁶³

D. Licensing of Software

Licensing serves as the dominant transactional form in the software industry.⁶⁴ Software licenses are contracts where licensors give licensees the right to use the software for a set period of time under certain conditions.⁶⁵ Such

Software Patents from Overly Broad Business Method Patents, 30 BERKELEY TECH. L.J. 807, 807 (2015).

52. *Id.*

53. *Id.*; see *Gottschalk v. Benson*, 409 U.S. 63, 71-73 (1972).

54. *Diamond v. Diehr*, 450 U.S. 175, 187-88 (1981).

55. See *infra* Parts II, III.

56. 35 U.S.C. § 154(a) (2018).

57. *United States v. Univis Lens Co.*, 316 U.S. 241, 250 (1942).

58. *Keeler v. Standard Folding-Bed Co.*, 157 U.S. 659, 661 (1895).

59. *Univis Lens*, 316 U.S. at 251.

60. NGUYEN ET AL., *supra* note 2, at 8.

61. *Id.*

62. *Id.*

63. *Minebea Co. v. Papst*, 444 F. Supp. 2d 68, 157 (D.D.C. 2006).

64. NGUYEN ET AL., *supra* note 2, at 511.

65. RUSTAD, *supra* note 1, § 1.05.

transactions are necessary due to the nature of software:

Software is unlike most products. Its value lies not in any physical structure, such as the CD on which it may be encoded, but rather in the ephemeral code or instructions themselves. The structure containing the software—whether a piece of plastic, or a network of wires and switches (such as the Internet) connected to a memory—is merely the mechanism for delivery from publisher to consumer. And unlike other copyrighted works, it is not visual or audio aesthetics of software that most reflect its value, it is the effects of its operation upon a machine, such as a processor (though these effects often reflect great creativity). Put another way, the software “product” is the code together with the associated rights to run it.⁶⁶

In copyright law, licensees differ from copyright owners in that licensees own a copy of the copyrighted work, and only retain rights associated with the work as laid out in the license agreement.⁶⁷ Licensing is predominantly used because it enables software developers to deliver to various users the same code, but with unique assortments of rights.⁶⁸ A software developer may grant one licensee the right to fix bugs, another the right to use the software, and another the right to distribute the software.⁶⁹ Licenses allow software developers to collaborate and share intellectual property with each other, which aids the software development process.⁷⁰ Additionally, licenses enable software developers to bring software products to market in various ways.⁷¹

Software licensing transactions consist of four main categories: “[1] licenses to build products; [2] licenses to create customer solutions; [3] licenses to distribute software; and [4] licenses that describe usage.”⁷² The first two categories are often referred to as upstream licenses and the latter two are often referred to as downstream licenses.⁷³ Upstream licenses consist of licenses to build products,⁷⁴ licenses to create solutions,⁷⁵ and source code licensing.⁷⁶ Downstream licenses consist of distribution of software,⁷⁷ licenses that describe

66. *Id.* (quoting Brief for Software & Information Industry Association (SIIA) as Amicus Curiae Supporting Appellee and Affirmance on Appeal, *MDY Indus., LLC v. Blizzard Entm’t, Inc.*, 629 F.3d 928 (9th Cir. 2009) (Nos. 09-15932, 09-16044)).

67. DAVID NIMMER, 2 NIMMER ON COPYRIGHT § 8.08 (2017).

68. NGUYEN ET AL., *supra* note 2, at 511.

69. *Id.*

70. Robert W. Gomulkiewicz, *Is the License Still the Product?*, 60 ARIZ. L. REV. 425 (2018).

71. *Id.*

72. NGUYEN ET AL., *supra* note 2, at 511.

73. *Id.*

74. GOMULKIEWICZ ET AL., *supra* note 5, at 307.

75. *Id.*

76. *Id.* at 308.

77. *Id.* at 310.

usage,⁷⁸ and end-user licenses.⁷⁹ Upstream licenses represent the apex of software licensing law because, while programmers often develop software from scratch, software developers primarily collaborate with others when working on software.⁸⁰ As a result, software developers use licenses to assign different rights among the various contributors.⁸¹ These rights include everything from sharing confidential source code with the licensee⁸² to licenses to distribute software.⁸³ Depending on the type of license granted, the license subjects the licensee to various restrictions.⁸⁴

Upstream licensing encompasses three categories of licenses: licenses to build products, licenses to create solutions, and source code licenses.⁸⁵ Licenses to build products allow software developers to collaborate with each other when creating software programs.⁸⁶ Software developers use these licenses to assign rights among various project contributors.⁸⁷ In this regard, software developers treat licenses as legal tools when building software programs.⁸⁸ In fact, larger software programs often consist of integrated digital content from various software developers and companies where software developers jointly own code rights.⁸⁹ Additionally, licenses to create solutions allow for the integration of different computer technologies.⁹⁰ A computer system typically contains components from various sources.⁹¹ For example, one computer may contain an Intel microprocessor, a Dell central processing unit, a Samsung monitor, and a Microsoft operating system coupled with a Logitech mouse, a Hewlett-Packard printer, and a mix of software applications.⁹² Finally, source code licensing allows software developers to share both confidential and non-confidential source code.⁹³

II. THE SUPREME COURT AND SOFTWARE INTELLECTUAL PROPERTY PROTECTION

The Supreme Court has struggled with the patentability of software for nearly

78. *Id.* at 311.

79. *Id.*

80. NGUYEN ET AL., *supra* note 2, at 512.

81. *Id.*

82. *Id.* at 514.

83. *Id.* at 532.

84. *See id.* at 537-45.

85. GOMULKIEWICZ ET AL., *supra* note 5, at 307-08.

86. *Id.* at 307.

87. *Id.*

88. *Id.*

89. *Id.*

90. *Id.*

91. *Id.*

92. *Id.*

93. *Id.* at 308.

forty years.⁹⁴ In the 1970s, the Court confronted the issue of software patentability for the first time with *Gottschalk v. Benson* and *Parker v. Flook*.⁹⁵ In these cases, the Court declined to extend patent protection to software.⁹⁶ In the 1980s, the Supreme Court lowered the patent-eligibility threshold for software with *Diamond v. Diehr*.⁹⁷ But after nearly thirty years of maintaining a low bar for patent-eligibility for software, the Supreme Court raised the threshold for software patent-eligibility again with *Bilski v. Kappos* and *Mayo Collaborative Services v. Prometheus Laboratories, Inc.*⁹⁸ This Note highlights the two more recent cases—*Alice* and *Lexmark*—in which the Supreme Court continued to—either directly or indirectly—impact software patenting.⁹⁹ The Supreme Court’s 2014 *Alice* decision directly impacted the patent-eligibility of software.¹⁰⁰ In *Alice*, the Supreme Court held that the method claims at issue consisted of an abstract idea, and that requiring computer implementation of the methods in the patent claims did not constitute a transformation of an “abstract idea into a patent-eligible invention.”¹⁰¹ Most recently in 2017, the Supreme Court decided *Lexmark*, where it clarified the patent exhaustion doctrine.¹⁰² While the full impact of *Lexmark* is yet to be seen, this Note predicts that it likely will have an indirect impact on the software industry and patenting software.

A. The Alice Decision

A steep decline in protecting software through patent law began in 2014 with the Supreme Court’s *Alice* decision.¹⁰³ Alice Corporation owned several patents, which claimed a computer-implemented method for controlling settlement risk, or “the risk that only one party to a financial transaction will pay what it owes.”¹⁰⁴ The method patents mitigated settlement risk by facilitating “the exchange of financial obligations between two parties by using a computer system as a third-party intermediary.”¹⁰⁵ CLS Bank initiated a suit against Alice Corporation, claiming that Alice Corporation’s patents were invalid and unenforceable.¹⁰⁶ The

94. See Zivojnovic, *supra* note 51, at 807.

95. See *id.* at 811; see also *Parker v. Flook*, 437 U.S. 584 (1978); *Gottschalk v. Benson*, 409 U.S. 63 (1972).

96. *Parker*, 437 U.S. 584; *Gottschalk*, 409 U.S. 63; Zivojnovic, *supra* note 51, at 811.

97. *Diamond v. Diehr*, 450 U.S. 175 (1981); Zivojnovic, *supra* note 51, at 814.

98. *Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, 566 U.S. 66 (2012); *Bilski v. Kappos*, 561 U.S. 593 (2010); Zivojnovic, *supra* note 51, at 816.

99. *Impression Prods. v. Lexmark Int’l, Inc.*, 137 S. Ct. 1523 (2017); *Alice Corp. v. CLS Bank Int’l*, 573 U.S. 208 (2014).

100. *Alice*, 573 U.S. at 212.

101. *Id.*

102. *Lexmark*, 137 S. Ct. at 1529.

103. *Alice*, 573 U.S. 208.

104. *Id.* at 212.

105. *Id.* at 213.

106. *Id.* at 214.

Court utilized the two-part framework set forth in *Mayo Collaborative Services v. Prometheus Laboratories, Inc.*¹⁰⁷ to decide whether the patent claims represented abstract ideas or “patent-eligible applications of those concepts.”¹⁰⁸ The *Mayo* framework required the Court to first examine whether the patent claims at issue directly corresponded to a patent-ineligible concept, such as an abstract idea.¹⁰⁹ Then, the framework required the Court to ascertain whether there were additional elements of the claim that transformed the claim into a patentable application of the patent-ineligible concept.¹¹⁰

The Supreme Court concluded that Alice Corporation’s patent claims at issue were directly correlated to a patent-ineligible concept, specifically “the abstract idea of intermediated settlement.”¹¹¹ The Court then turned to the second step of the *Mayo* framework and decided that the “method claims, which merely require[d] generic computer implementation, fail[ed] to transform that abstract idea into a patent-eligible invention.”¹¹² With this holding, the Court sent a message to the software industry: The simple addition of a computer system to an abstract idea, such as a system or method, fails to meet the subject matter patentability requirement.¹¹³

B. The Lexmark Decision

In *Lexmark*, the Supreme Court clarified the Patent Act’s patent exhaustion doctrine, holding that when a patent owner sells his or her patented product, the owner relinquishes all rights associated with that product, including any restrictions the patentee wishes to impose on the purchaser.¹¹⁴ Therefore, the Supreme Court expanded the application of the patent exhaustion doctrine to include all sales—not just those that are unrestricted and unconditional.¹¹⁵

Lexmark International brought an action against Impression Products for patent infringement.¹¹⁶ Lexmark designs, manufactures, and sells toner cartridges both in the United States and around the world.¹¹⁷ These cartridges can be refilled and reused when they run out of toner.¹¹⁸ Impression Products retrieved the empty Lexmark cartridges from consumers, refilled them with toner, and resold them to consumers at a lower price than what Lexmark sells cartridges for.¹¹⁹ In response,

107. *Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, 566 U.S. 66 (2012).

108. *Alice*, 573 U.S. at 217.

109. *Mayo*, 566 U.S. at 77-78.

110. *Id.* at 78.

111. *Alice*, 573 U.S. at 218.

112. *Id.* at 221.

113. *Id.* at 224.

114. *Impression Prods. v. Lexmark Int’l, Inc.*, 137 S. Ct. 1523, 1529 (2017).

115. *Id.* at 1535.

116. *Id.* at 1523.

117. *Id.* at 1529.

118. *Id.*

119. *Id.*

Lexmark encouraged customers to return the empty cartridges to Lexmark by giving them two options: Buy the cartridge at full price, or buy the cartridge at a discounted price through Lexmark's "Return Program."¹²⁰ To enforce the "Return Program" sales, Lexmark installed microchips on the cartridges, which prevented reuse of the toner cartridges.¹²¹ However, Impression Products continued acquiring empty cartridges and circumvented the effects of the microchips and continued to resell the toner cartridges.¹²² Consequently, Lexmark sued for patent infringement.¹²³ The Supreme Court found that once Lexmark sold the toner cartridges, it exhausted all associated patent rights.¹²⁴ Specifically, "[t]he single-use/no-resale restrictions in Lexmark's contracts with customers may have been clear and enforceable under contract law, but they do not entitle Lexmark to retain patent rights in an item that it has elected to sell."¹²⁵

In its decision, the Court distinguished between licensing and selling a product.¹²⁶ Licensing requires the patentee to exchange rights, where selling requires the patentee to exchange goods.¹²⁷ The Court clarified that patent exhaustion hinges on the "principle that, when an item passes into commerce, it should not be shaded by a legal cloud on title as it moves through the marketplace."¹²⁸ Therefore, because a license does not involve the passing of title to a product, a patentee can impose restrictions on licensees.¹²⁹ However, the patentee's ability to impose restrictions on licensees does not mean that "patentees can use licenses to impose post-sale restrictions on *purchasers* that are enforceable through the patent laws."¹³⁰ Instead, so long as the licensee follows the conditions of the license when selling the product, the licensee's sale is treated as if the patentee made the sale.¹³¹ Consequently, the sale exhausts the patentee's patent rights.¹³² As a result, the Court held that "[o]nce a patentee decides to sell—whether on its own or through a licensee—that sale exhausts its patent rights, regardless of any post-sale restrictions the patentee purports to impose, either directly or through a license."¹³³

120. *Id.* at 1530.

121. *Id.*

122. *Id.*

123. *Id.*

124. *Id.* at 1531.

125. *Id.*

126. *Id.* at 1534.

127. *Id.*

128. *Id.*

129. *Id.*

130. *Id.* at 1534-35 (emphasis in original).

131. *Id.* at 1535.

132. *Id.*

133. *Id.*

III. THE AFTERMATH OF THE *ALICE* DECISION

A. The Alice Decision's Impact on Patents

One week after the *Alice* decision, the United States Patent and Trademark Office issued a guidance memorandum to help patent examiners in the implementation of *Alice*.¹³⁴ According to the memorandum, any claim implementing an abstract idea must amount to “significantly more” than the abstract idea alone.¹³⁵ For example, the memorandum stated that software claims would more likely meet the requirement if the claims contain either “[i]mprovements to the functioning of the computer itself; [or m]eaningful limitations beyond generally linking the use of an abstract idea to a particular technological environment.”¹³⁶

In the aftermath of the United States Patent and Trademark Office memorandum and within two months of the *Alice* decision, 830 patent applications were withdrawn from the United States Patent and Trademark Office.¹³⁷ Within one year of the *Alice* decision, 286 patents were invalidated in either the district court, the federal circuit, or by Patent Trial and Appeal Board decisions.¹³⁸ In fact, out of 196 patent applications, the Patent Trial and Appeal Board upheld only 18 patent applications and invalidated 178 applications.¹³⁹ Furthermore, the district courts maintained a patent invalidation rate of 69.7%, and the Federal Circuit maintained a patent invalidation rate of 94.1%.¹⁴⁰ Within the first two years after the *Alice* decision, the approval and upholding of software patents continued to decline.¹⁴¹ The Federal Circuit maintained an average patent invalidation rate of 91.9%, with a slight, almost unnoticeable, drop in its patent invalidation rate from the year before.¹⁴² The United States Patent and Trademark Office rejected more than 36,000 patent applications, and applicants abandoned over 5,000 patent applications.¹⁴³

B. The Alice Decision's Impact on the Software Industry

Though the *Alice* decision's reach is not necessarily targeted in scope, the decision particularly affected the software industry.¹⁴⁴ As of January 2018, the

134. See Memorandum from Andrew H. Hirshfeld, Deputy Comm'r of the U.S. Patent & Trademark Office, to Patent Examiners (June 25, 2014), available at https://www.uspto.gov/sites/default/files/patents/announce/alice_pec_25jun2014.pdf [<https://perma.cc/3T2R-YK9C>].

135. *Id.*

136. *Id.*

137. *One Year after Alice*, *supra* note 13.

138. *Id.* at 534.

139. *Id.* at 540.

140. *Id.*

141. See *Two Years after Alice*, *supra* note 13.

142. *Id.* at 358.

143. *Id.* at 358-359.

144. See *One Year after Alice*, *supra* note 13; see also *Two Years after Alice*, *supra* note 13,

Federal Circuit issued just six opinions in which it upheld the patentability of a software invention in light of the *Alice* decision: (1) *DDR Holdings, LLC v. Hotels.com*¹⁴⁵ (Dec. 5, 2014); (2) *L.P., Enfish, LLC v. Microsoft Corp.*¹⁴⁶ (May 12, 2016); (3) *BASCOM Global Internet Services v. AT&T Mobility LLC*¹⁴⁷ (June 27, 2016); (4) *McRO, Inc. v. Bandai Namco Games*¹⁴⁸ (Sept. 13, 2016); (5) *Amdocs (Isr.) Ltd. v. Openet Telecom, Inc.*¹⁴⁹ (Nov. 1, 2016); and (6) *Thales Visionix v. United States*¹⁵⁰ (Mar. 8, 2017). Out of these six decisions, *DDR Holdings* was the only decision where implementation of an abstract idea amounted to an inventive concept that rendered the computer-implemented invention patent-eligible.¹⁵¹

In *Enfish*, the Federal Circuit upheld the patentability of a computer-implemented invention under *Alice*, reasoning “the plain focus of the claims is on an improvement to computer functionality itself, not on economic or other tasks for which a computer is used in its ordinary capacity.”¹⁵² In this decision, the Federal Circuit reframed the first step in *Alice*’s two-step analysis, transforming the step into a “bright-line dichotomy.”¹⁵³ The court asked “whether the focus of the claims is on the specific asserted improvement in computer capabilities . . . or, instead, on a process that qualifies as an ‘abstract idea’ for which computers are invoked merely as a tool.”¹⁵⁴ Put simply, the *Enfish* framework inquires whether “the claimed invention [is] something that makes a computer work better? Or are computers merely being used to do another task?”¹⁵⁵ If the claimed invention enables the computer to work better, it is patentable.¹⁵⁶ If the claimed invention merely uses a computer to complete another task, the court proceeds to the second step of the *Alice* analysis.¹⁵⁷ After the *Enfish* decision, the United States Patent and Trademark Office again issued a new guidance opinion instructing United States Patent and Trademark Office Examiners to adhere to *Enfish*’s framework for the first step of the 2-part *Alice* analysis.¹⁵⁸

at 358-359.

145. *DDR Holdings, LLC v. Hotels.com, L.P.*, 773 F.3d 1245 (Fed. Cir. 2014).

146. *Enfish, LLC v. Microsoft Corp.*, 822 F.3d 1327 (Fed. Cir. 2016).

147. *BASCOM Global Internet Servs. v. AT&T Mobility LLC*, 827 F.3d 1341 (Fed. Cir. 2016).

148. *McRO, Inc. v. Bandai Namco Games*, 837 F.3d 1299 (Fed. Cir. 2016).

149. *Amdocs (Isr.) Ltd. v. Openet Telecom, Inc.*, 841 F.3d 1288 (Fed. Cir. 2016).

150. *Thales Visionix v. United States*, 850 F.3d 1343 (Fed. Cir. 2016).

151. *DDR Holdings, LLC v. Hotels.com, L.P.*, 773 F.3d 1245, 1259 (Fed. Cir. 2014).

152. *Enfish, LLC v. Microsoft Corp.*, 822 F.3d 1327, 1336 (Fed. Cir. 2016).

153. *Two Years after Alice*, *supra* note 13, at 365.

154. *Enfish*, 822 F.3d at 1335-36.

155. Jason Rantanen, *Judge Hughes and the New § 101 Dichotomy*, PATENTLY-O (May 23, 2016), <http://patentlyo.com/patent/2016/05/hughes-%C2%A7-dichotomy.html> [<https://perma.cc/D5QV-YBDR>].

156. *Id.*

157. *Id.*

158. Memorandum from Robert W. Bahr, Deputy Comm’r for Patent Examining Policy of the

In the aftermath of *Alice*, the software industry has had to navigate the wavering boundaries of patent protection for its computer-implemented inventions.¹⁵⁹ Some scholars have written articles on proposed policy reform for patenting software,¹⁶⁰ and others have written advice on navigating patenting software in a post-*Alice* world.¹⁶¹ But the decision still stands in the way of patenting many computer-implemented inventions and software.¹⁶²

IV. THE *LEXMARK* DECISION'S IMPACT ON THE SOFTWARE INDUSTRY

A. *Conflicting Interpretations of the Lexmark Decision*

The *Lexmark* decision has also led to uncertainty regarding its interpretation. Although the Supreme Court's goal of strengthening the patent exhaustion doctrine lightened some of the "legal cloud cast by patent rights," the extent of the decision's impact remains unclear.¹⁶³ For instance, the Court focuses on the idea that items should pass through commerce free of restraints, but patents nevertheless still provide the restraints the Court sought to limit, and patents will continue to do so despite the Court's ruling in *Lexmark*. Additionally, the impact of the *Lexmark* decision may not be as absolute as the Court intended because "there is generally no guarantee that all relevant patentees have sold (or authorized the sale of) all components to be found within any given item in commerce."¹⁶⁴ To illustrate this point, take the Court's own illustration of a car-repair shop that repairs and resells used cars.¹⁶⁵ The Supreme Court explains that the "shop can rest assured that, so long as those bringing in the cars own them, the shop is free to repair and resell those vehicles."¹⁶⁶ But, to be completely sure, "the shop would also need to know that all owners of all patents covering any

U.S. Patent & Trademark Office, to the Patent Examining Corps (May 19, 2016), *available at* https://www.uspto.gov/sites/default/files/documents/ieg-may-2016_enfish_memo.pdf [<https://perma.cc/VCK9-KSF3>].

159. See Gene Quinn, *How to Patent Software in a Post Alice Era*, IPWATCHDOG (Nov. 17, 2016), <http://www.ipwatchdog.com/2016/11/17/patent-software-post-alice/id=74750/> [<https://perma.cc/3XKQ-76BV>]; see also Paul J. Sutton, *Patent Eligibility After Alice*, WORLD INTELL. PROP. REV. (Feb. 27, 2017), <https://www.worldipreview.com/contributed-article/patent-eligibility-after-alice> [<https://perma.cc/NY8D-KUPY>].

160. See Colleen V. Chien, *Reforming Software Patents*, 50 HOUS. L. REV. 325 (2012); Charles F. Green, *CLS Bank v. Alice Corp.: What Does It Mean for Software Patent Eligibility?*, 13 J. MARSHALL REV. INTELL. PROP. L. 601 (2014).

161. See Sutton, *supra* note 159; see also Quinn, *supra* note 159.

162. See Quinn, *supra* note 159.

163. Andrew C. Michaels, *Impression Products, Inc. v. Lexmark International, Inc.: A Glib Rebuke of the Federal Circuit*, GEO. WASH. L. REV. ON THE DOCKET (June 11, 2017), <http://www.gwlr.org/impression-products-v-lexmark/> [<https://perma.cc/3BFG-D8F7>].

164. *Id.*

165. *Impression Prods., Inc. v. Lexmark Int'l, Inc.*, 137 S. Ct. 1523, 1532 (2017).

166. *Id.*

component of the car have sold or authorized a sale of that component.”¹⁶⁷ This poses a significant challenge, because even a “generic smartphone assembled from various high-tech components could practice an estimated 250,000 patents,” owned by various persons and entities.¹⁶⁸

Moreover, the *Lexmark* decision arguably neglects to consider the heightened standard of exhaustion in relation to licenses to distribute.¹⁶⁹ Specifically, infringement can still occur, even after the first sale of the patented item, if the sale was unauthorized.¹⁷⁰ A licensee makes an unauthorized sale if the license restricts the licensee’s authority to sell and he “knows that the purchaser plans to use the item in prohibited ways.”¹⁷¹ Therefore, a purchaser who purchases the item from a restricted licensee and uses the item in a way that is inconsistent with the license restrictions lacks authority to do so and infringes on the patent.¹⁷²

An additional consequence of the *Lexmark* decision is that businesses may pursue other methods for price discrimination and control of sales.¹⁷³ Licensing is one possible method, as the *Lexmark* decision may lead to fewer sales and more licensing.¹⁷⁴ However, licensing generally goes against the law’s aversion to restraints on alienation.¹⁷⁵ And licensing “was on the rise before this case and would be just as desirable (to companies and some consumers) or undesirable (to other consumers and resellers) for goods not covered by patents as for those that are.”¹⁷⁶ In fact, the *Lexmark* decision “opens the door for creative contract lawyers to draft licensing agreements that severely restrict resale of patented products.”¹⁷⁷ Companies can potentially replace sales with licenses to bypass the *Lexmark* decision. To this end, what if *Lexmark* had replaced the terms in its agreements, “so that instead of ‘selling’ cartridges subject to a resale restriction, it ‘licenses’ the cartridges to the customers for an indefinite term on the condition that they not resell the cartridges to anyone else?”¹⁷⁸ Under this theory, it appears that the grant of a license, rather than a sale, would enable *Lexmark* to retain its

167. Michaels, *supra* note 163.

168. *Id.* (quoting *Lexmark*, 137 S. Ct. at 1532).

169. *Id.*

170. *Id.*

171. *Id.*

172. *Id.*

173. Dennis Crouch, *Impression Products, Inc. v. Lexmark Inc.: Will International Patent Exhaustion Bring Free Trade in Patented Goods?*, PATENTLY-O (June 1, 2017), https://patentlyo.com/patent/2017/06/impression-international-exhaustion.html#_ftn1 [<https://perma.cc/W28P-VGPZ>].

174. *Id.*

175. *Id.*

176. *Id.*

177. Daniel Hemel & Lisa Larrimore Ouellette, *Licensing in the Shadow of Impression Products*, WRITTEN DESCRIPTION (May 31, 2017, 12:31 AM), <https://writtendescription.blogspot.com/2017/05/licensing-in-shadow-of-impression.html> [<https://perma.cc/7687-C6TS>].

178. *Id.*

patent rights.¹⁷⁹ On the other hand, courts may decide in the future that licenses are essentially sales in disguise, questioning the difference between “(1) a sale on the condition that the purchaser cannot resell the product and (2) a license that allows the licensee to do whatever she wants with the product except resell it.”¹⁸⁰ If such licenses were found to be sales, the *Lexmark* decision would supply the applicable rule of law.¹⁸¹

B. The Lexmark Decision’s Impact on Software Licensing

With the *Lexmark* decision, the Supreme Court laid down an absolute: When a patent owner sells his or her patented product, the owner relinquishes all rights associated with that product, including restrictions the patentee tries to impose on the purchaser.¹⁸² This absolute presents an uncompromising bright-line rule that applies to all patent holders, regardless of what type of patent they own or to which industry their patent belongs.¹⁸³ But software is a bit of a gray area in intellectual property law.¹⁸⁴ Software companies generally transact in the form of licenses, and the Court makes a distinction about *Lexmark*’s application to licenses.¹⁸⁵ The Court specifies that patentees can impose restrictions on licensees because a license does not involve the passing of title to a product, though patentees cannot utilize licenses as a means of imposing post-sale restrictions on purchasers enforceable through patent law.¹⁸⁶ But software licensing is more complicated than what the Court discusses in the *Lexmark* decision. The software provides functionality of the application—what the company *can* do with the software. The license provides the terms of the software—what the company *may* do with the software.¹⁸⁷ In fact, the Seventh Circuit recognizes that when dealing with software, “[i]n the end, the terms of the license are conceptually identical to the contents of the package.”¹⁸⁸ Essentially, the license is the product.¹⁸⁹

The best way to highlight the impact *Lexmark*, or lack thereof, is to consider

179. *Id.*

180. *Id.*

181. *Id.*

182. *Impression Prods., Inc. v. Lexmark Int’l, Inc.*, 137 S. Ct. 1523, 1529 (2017).

183. Samuel McMahon, *Supreme Court Announces ‘Uniform and Automatic’ Rule for Patent Exhaustion*, IP INTELLIGENCE (June 1, 2017), <https://www.ipintelligencereport.com/2017/06/01/supreme-court-announces-uniform-and-automatic-rule-for-patent-exhaustion/> [<https://perma.cc/X7ZR-WSFT>].

184. Andrés Guadamuz González, *Software Patentability: Emerging Legal Issues*, WIPO MAG. (Dec. 2008), http://www.wipo.int/wipo_magazine/en/2008/06/article_0006.html [<https://perma.cc/T427-L88Q>].

185. *Lexmark*, 137 S. Ct. at 1534.

186. *Id.* at 1534-35.

187. See Robert W. Gomulkiewicz, *The License Is the Product: Comments on the Promise of Article 2B for Software and Information Licensing*, 13 BERKELEY TECH. L.J. 891 (1998).

188. *ProCD, Inc. v. Zeidenberg*, 86 F.3d 1447, 1453 (7th Cir. 1996).

189. *Id.*

several scenarios in which the *Lexmark* decision would come into play.

1. *Scenario 1.*—Software Developer is a company that creates patented software. Smartphone Producer is a company that creates mobile phones. Software Developer licenses its software to Smartphone Producer for use in Smartphone Producer's mobile phones. The license is a limited license that authorizes Smartphone Producer to use Software Developer's software to manufacture, market, sell, and distribute the phones. The license prohibits Smartphone Producer from allowing anyone other than Smartphone Producer to use Software Developer's software. Overwhelmed with the amount of work to be done, Smartphone Producer decides to outsource some of its production to Overseas Manufacturer. Such outsourcing requires Smartphone Producer to provide Overseas Manufacturer with Software Developer's software, so Smartphone Producer licenses the software to Overseas Manufacturer. Overseas Manufacturer uses Software Developer's software to complete its outsourcing work. Software Developer sues Smartphone Producer for patent infringement claiming that Smartphone Producer did not have permission to license Software Developer's software with Overseas Manufacturer, because the license limited Smartphone Producer's ability to use the software in any way inconsistent with manufacture, market, sale, and distribution.

In this scenario, because Smartphone Producer did not adhere to the license agreement, Software Developer could bring a patent infringement suit against Smartphone Producer. Prior to the *Lexmark* decision, this conduct would be infringing because the license between Software Developer and Smartphone Producer imposed restrictions on Smartphone Producer, and the license needed to be unconditional and unrestricted in order for patent exhaustion to apply.¹⁹⁰ Under the *Lexmark* ruling, patent exhaustion only applies to licensees when the licensee adheres to the conditions and limitations of the license.¹⁹¹ However, there is some gray area here because the Court's reasoning relies on the assumption that licensing only involves the passage of rights rather than the product.¹⁹² But when dealing with software, courts view the license as the product.¹⁹³ In such a case, if a court treated the license as the product, then the transaction between Software Developer and Smartphone Producer constitutes a sale, which would exhaust The Software Developer's rights associated with the patented software.

2. *Scenario 2.*—Software Developer is a company that creates patented software. Fitness Device Producer is a company that creates fitness tracking devices. Software Developer licenses its software to Fitness Device Producer for use in Fitness Device Producer's fitness tracking devices. The license is a limited license that authorizes Fitness Device Producer to use Software Developer's software to manufacture, market, sell, and distribute the fitness tracking devices. The license also grants Fitness Device Producer the right to take any foreseeable measure to aid in the manufacture, market, sale, and distribution of the fitness

190. NGUYEN ET AL., *supra* note 2, at 8.

191. *Lexmark*, 137 S. Ct. at 1535.

192. *Id.* at 1534.

193. *ProCD*, 86 F.3d at 1453.

tracker in relation to Software Developer's software. Fitness Device Producer decides to outsource some of its production to Manufacturer. Such outsourcing requires Fitness Device Producer to provide Manufacturer with Software Developer's software, which both Software Developer and Fitness Device Producer agree is foreseeable. Fitness Device Producer licenses Software Developer's software to Manufacturer in an unrestricted license. Manufacturer uses Software Developer's software to complete their outsourcing work. Manufacturer then also uses Software Developer's software for a different project. Software Developer sues Manufacturer for patent infringement, claiming that Manufacturer lacked permission to use Software Developer's software.

This scenario presents a different issue: Fitness Device Producer followed the terms of the limited license provided by Software Developer. The Court determined that patent exhaustion applies to licensees when the licensee adheres to the conditions and limitations of the license.¹⁹⁴ Because Fitness Device Producer adhered to the conditions and limitations of the license, the transaction between Fitness Device Producer and Manufacturer exhausts Software Developer's rights in the patented software. Fitness Device Producer had permission to take any foreseeable measure to aide in the manufacture, market, sale, and distribution of the fitness tracker in relation to Software Developer's software. Licensing Software Developer's software to Manufacturer fit into this provision, though perhaps Fitness Device Producer should not have given Manufacturer an unrestricted license. So, the transaction between Fitness Device Producer and Manufacturer served to exhaust Software Developer's rights in the patented software. As a result, Software Developer should not win in an infringement lawsuit against Manufacturer, according to the Court's logic and reasoning.

C. The Lexmark Decision's Impact on Software Patenting

Software developers create software through an iterative process of collaborating and sharing ideas.¹⁹⁵ Software companies license their products to collaborate and share intellectual property in the industry.¹⁹⁶ Collaboration and cooperation between software developers working at different organizations in different geographic regions often lead to the creation of software that people depend upon every day.¹⁹⁷ Additionally, increased innovation is one of the main purposes of patent law.¹⁹⁸ The way the software industry modeled its licensing transactional process furthers the innovational goal of patent law by allowing software developers and organizations to share their ideas and creations without

194. *Lexmark*, 137 S. Ct. at 1535.

195. See Gomulkiewicz, *supra* note 70; see also Robert E. Kraut & Lynn A. Streeter, *Coordination in Software Development*, COMM. ACM, Mar. 1995, at 70.

196. See Gomulkiewicz, *supra* note 70.

197. Kraut & Streeter, *supra* note 195.

198. *Lexmark*, 137 S. Ct. at 1532.

relinquishing their rights associated with inventions.¹⁹⁹ However, the *Lexmark* decision will disrupt this process by discouraging patent owners from licensing software because once they do, they may lose protection.²⁰⁰ Consider the scenarios above. While the *Lexmark* decision does not necessarily resolve some of the gray areas presented, it leaves a precedent that appears to disfavor the collaborative nature of software development. In turn, this impedes the main purpose of patent law, to promote the growth of science and technological innovation.

V. THE FUTURE OF PROTECTING SOFTWARE

Software is intellectual property, governed by both copyright²⁰¹ and patent law.²⁰² Copyright law protects software, but typically provides a thin shield in comparison to patent law because unlike patent law, copyright law does not protect functionality.²⁰³ Rather, copyright law protects the written code, and sometimes the structure and layout of the code.²⁰⁴ Developers can easily mimic functionality of software as there are countless ways to write the code to achieve the same result.²⁰⁵ For example, two software developers can write two different software programs, using different programming languages, that create the same functionality.²⁰⁶ As a result, protecting the underlying functionality of software is crucial and arguably more important than solely protecting the written code.

Consequently, inventors use patent law to protect functionality.²⁰⁷ Although patent law is the facet of intellectual property law designed to protect functionality in inventions and functional products,²⁰⁸ the *Alice* decision made patenting software and computer-implemented inventions much more difficult,²⁰⁹

199. Gomulkiewicz, *supra* note 70, at 1; see Kraut & Streeter, *supra* note 195.

200. Mark Baghdassarian & Fredrich Laub, *Patent Exhaustion Ruling Will Affect Many Industries*, LAW360 (June 26, 2017, 1:30 PM), <https://www.law360.com/articles/935945/patent-exhaustion-ruling-will-affect-many-industries>.

201. See *Apple Comput., Inc. v. Microsoft Corp.*, 35 F.3d 1435 (9th Cir. 1994); *Apple Comput., Inc. v. Franklin Comput. Corp.*, 714 F.2d 1240 (3d Cir. 1983).

202. NGUYEN ET AL., *supra* note 2, at 507.

203. James Straatman, *The Difference Between Copyright and Patent Protections on Software*, PLAGIARISM TODAY (May 14, 2015), <https://www.plagiarismtoday.com/2015/05/14/the-difference-between-copyright-and-patent-protections-on-software/> [https://perma.cc/A4YH-5RVG].

204. *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339 (Fed. Cir. 2014); *Johnson Controls, Inc. v. Phoenix Control Systems, Inc.*, 866 F.2d 1173, 1175 (9th Cir. 1989).

205. See, e.g., *Say 'Hello World' in 28 Different Programming Languages*, EXCEL WITH BUS., <https://excelwithbusiness.com/blog/say-hello-world-in-28-different-programming-languages/> [https://perma.cc/D36K-LUH9] (last visited May 28, 2019).

206. See, e.g., *id.*

207. 35 U.S.C. § 101 (2018).

208. *Id.*

209. See *Alice Corp. v. CLS Bank Int'l*, 573 U.S. 208, 220-22 (2014); *Two Years after Alice*, *supra* note 13.

and the *Lexmark* decision may further dissuade software companies from seeking patent protection if protection ceases upon the first sale of the product.²¹⁰ Because the functionality behind the software is what needs protecting, the lessened availability of patent protection presents huge issues to the software industry in how it protects its products. Considering the patentability limitations from the recent Supreme Court decisions, how can software companies ensure that the functionality behind their intellectual property is protected? Can another branch of intellectual property achieve the same, or similar, protection to that of patent law? This Note presents two possible solutions: (1) creatively utilizing copyright law to achieve protection similar to that normally provided by patent law; or (2) creating a *sui generis* system of intellectual property law to specifically govern and protect software.

A. Copyright Protection of Software

Copyright law can be used to protect the functionality of software, but only in some capacity because by definition computer programs are functional, as “they are designed to accomplish some task.”²¹¹ In fact, the Copyright Act defines computer programs as “a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result.”²¹² Consequently, courts have held that an original work, even if functional, “is entitled to copyright protection as long as the author had multiple ways to express the underlying idea.”²¹³ Courts rationalize this exception to the functionality doctrine by reasoning that withholding copyright protection from an original work because it also carries out some function would preclude all computer programs from copyright protection, which goes against what Congress intended when it expressly afforded copyright protection to computer programs.²¹⁴

In the 2014 case of *Oracle America, Inc. v. Google, Inc.*, the Federal Circuit provided the best example for how copyright law can be used to protect software.²¹⁵ Oracle owns the copyright on the Java platform and all corresponding application programming interface (API) packages.²¹⁶ As of 2008, the Java

210. See Gomulkiewicz, *supra* note 70, at 4.

211. *Oracle Am., Inc. v. Google, Inc.*, 750 F.3d 1339, 1367 (Fed. Cir. 2014).

212. 17 U.S.C. § 101 (2018).

213. *Oracle*, 750 F.3d at 1367.

214. H.R. Rep. No. 94-1476, at 54 (1976); *Oracle*, 750 F.3d at 1367.

215. *Oracle*, 750 F.3d 1339.

216. *Id.* Java is a human-readable programming language, source code, that gets converted into binary or machine language, object code, which the computer is able to read. *Id.* at 1348. The API packages are pre-written programs that perform various common functions and enable software developers to use the prewritten code to build the common functions into their own programs, as opposed to each individual developer writing the common programs from scratch. *Id.* at 1349. See also *Classes and Objects in Java*, GEEKSFORGEEKS, <https://www.geeksforgeeks.org/classes-objects-java/> [<https://perma.cc/Y5ZH-ZVGF>] (explaining the basics of object oriented programming).

platform consisted of “more than 6,000 methods making up more than 600 classes grouped into 166 API packages.”²¹⁷ These packages contain “thousands of individual elements, including classes, subclasses, methods, and interfaces.”²¹⁸ Each package consists of two types of source code: declaring code and implementing code.²¹⁹ Declaring code identifies and explains the functionality of the implementing code.²²⁰ Implementing code provides the computer with a set of instructions for executing the function declared by the declaring code.²²¹ In *Oracle*, the Federal Circuit found the declaring code—as well as the structure and organization of the Java API packages—to be original.²²² Additionally, the Federal Circuit reasoned that though short phrases cannot be copyrighted, “an original combination of elements can be copyrightable.”²²³ Oracle sought protection not for individual short phrases, or lines of code, but for thousands of lines of code in which it “‘exercised creativity in the selection and arrangement’ of the method declarations when it created the API packages and wrote the relevant declaring code.”²²⁴ Consequently, the Federal Circuit held such creative expression worthy of copyright protection.²²⁵ As a result, Oracle obtained protection in a way not currently possible with patent law.

But extending copyright protection over entire programming language platforms and APIs may be going too far.²²⁶ Allowing companies to copyright entire APIs goes against the idea of code sharing, which is the foundational principle of software development and essential to the software development process.²²⁷ In fact, APIs are more than just packages of code; they allow different computer programs to communicate.²²⁸ If one party holds a copyright in an

217. *Id.* at 1349.

218. *Id.*

219. *Id.*

220. *Id.*

221. *Id.*

222. *Id.* at 1356.

223. *Id.* at 1362.

224. *Id.* at 1363.

225. *Id.*

226. See Venkatesh Ganesh, *Oracle’s Victory Over Google on API Copyright May Impact Software Development*, HINDU BUS. LINE (Nov. 25, 2017), <http://www.thehindubusinessline.com/info-tech/Oracle%E2%80%99s-victory-over-Google-on-API-copyright-may-impact-software-development/article20778996.ece> [<https://perma.cc/H6YZ-CNTZ>]; see also Julie Samuels, *Oracle v. Google and the Dangerous Implications of Treating APIs as Copyrightable*, ELECTRONIC FRONTIER FOUND. (May 7, 2012), <https://www.eff.org/deeplinks/2012/05/oracle-v-google-and-dangerous-implications-treating-apis-copyrightable> [<https://perma.cc/X9UZ-4WZW>]; Steven J. Vaughan-Nichols, *Oracle v. Google, and the End of Programming as We Know It*, COMPUTERWORLD (May 16, 2016 9:27 AM), <https://www.computerworld.com/article/3070001/application-development/oracle-v-google-and-the-end-of-programming-as-we-know-it.html> [<https://perma.cc/9R43-568S>].

227. See Ganesh, *supra* note 226.

228. Vaughan-Nichols, *supra* note 226.

API—thereby maintaining control over the API—that party “can determine who can make compatible and interoperable software.”²²⁹ Put simply, giving one party control over an API allows that party to “control add-on software development for that platform.”²³⁰ This would give larger software companies the ability to screen what applications can run on their platforms.²³¹ As a result, allowing APIs to be copyrighted inhibits the interoperability of programs and, in turn, hinders the innovation that copyright law seeks to promote.²³²

B. Creation of a Special Form of Intellectual Property Law for Software

Although copyright law can currently provide protection for software, copyright law’s limitations indicate that software may need an intellectual property law category of its own. From time to time, Congress recognizes that existing intellectual property law does not adequately cover all innovations and inventions.²³³ *Sui generis* systems involve the creation of separate law for subjects that lack requirements necessary to establish either copyright or patent protection.²³⁴ A *sui generis* system simply refers to one that is “of its own kind.”²³⁵ In such cases, Congress creates special forms of intellectual property to cover certain industries or subjects.²³⁶ For example, Congress passed the 1984 Semiconductor Chip Protection Act (SCPA) to establish a special form of intellectual property protection for technology related to semiconductor chips.²³⁷ Before the SCPA passed, semiconductor chip firms sought copyright protection for the semiconductor chips.²³⁸ But the Copyright Office continuously denied copyright protection to semiconductor chips due to their non-copyright-eligible utilitarian nature²³⁹ and Congress declined to extend copyright protection to semiconductor chips. Instead, Congress passed the SCPA to form a specialized intellectual property protection for the chips.²⁴⁰

229. Samuels, *supra* note 226.

230. *Id.*

231. *See id.*

232. Samuels, *supra* note 226; Vaughan-Nichols, *supra* note 226.

233. *See* J.H. Reichman, *Legal Hybrids Between the Patent and Copyright Paradigms*, 93 COLUM. L. REV. 2332 (1994).

234. J.H. Reichman & Pamela Samuelson, *Intellectual Property Rights in Data?*, 50 VAND. L. REV. 51, 53 (1997).

235. *Sui Generis*, BLACK’S LAW DICTIONARY (10th ed. 2014).

236. Reichman & Samuelson, *supra* note 234, at 53; *see also* Six “Other” Types of Intellectual Property, DUNLAP, BENNETT & LUDWIG, <https://www.dblawyers.com/intellectual-property/six-types-intellectual-property/> [<https://perma.cc/5HFG-AQ7W>] (last visited on Mar. 13, 2018).

237. DONALD S. CHISUM ET AL., UNDERSTANDING INTELLECTUAL PROPERTY LAW 738 (3d ed. 1992).

238. Pamela Samuelson, *Creating a New Kind of Intellectual Property: Applying the Lessons of the Chip Law to Computer Programs*, 70 MINN. L. REV. 471, 476 (1985).

239. *Id.* at 477; 17 U.S.C. §§ 101-102.

240. CHISUM ET AL., *supra* note 237, at 738; *see* Samuelson, *supra* note 238, at 477-507.

Like semiconductor chips, software does not truly meet the requirements for copyright protection.²⁴¹ Copyright law explicitly excludes functional or useful articles from protection.²⁴² But computer programs consist of integrated methods of instructions, specifically designed to carry out a function.²⁴³ Courts have acknowledged this and have chosen to uphold the copyright eligibility of the instructions despite their functional nature.²⁴⁴ But the fact stands that copyright law unambiguously excludes useful or functional articles from protection, and computer programs are—at their core—functional in nature.²⁴⁵ Furthermore, even though copyright law protects computer programs despite their functional nature, copyright law does not protect the functionality itself.²⁴⁶

Additionally, software does not easily meet the requirements for patent protection.²⁴⁷ Over the past forty years, the Supreme Court has struggled with the patentability of software.²⁴⁸ In *Alice*, the Court raised the threshold for software patent eligibility by holding that the simple addition of a computer system to an abstract idea fails to meet the subject matter patentability requirement.²⁴⁹ But many software programs consist of abstract ideas implemented using computer systems and are not eligible for patent protection as a result.²⁵⁰ Furthermore, even if a software program does meet the patentability requirements, the *Lexmark* decision provides an additional hurdle.²⁵¹ Software companies may be dissuaded from seeking patent protection if such protection ceases upon the first sale of the product.²⁵²

For the foregoing reasons, creation of a special form of intellectual property law to govern software may be the best answer for protecting software in a world where neither copyright nor patent law provides adequate protection. Congress should choose to create a *sui generis* system for the protection of software intellectual property. Such protection could be different in nature from that of patent and copyright. For example, the duration of protection for software could

241. 17 U.S.C. §§ 101-102; Samuelson, *supra* note 238, at 477.

242. 17 U.S.C. §§ 101-102.

243. *See Apple Comput., Inc. v. Franklin Comput. Corp.*, 714 F.2d 1240, 1251 (3d Cir. 1983).

244. *See id.*

245. *See* 17 U.S.C. §§ 101-102; *Franklin*, 714 F.2d at 1251.

246. *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1355-1356 (Fed. Cir. 2014); *Johnson Controls, Inc. v. Phoenix Control Systems*, 886 F.2d 1173, 1175 (9th Cir. 1989); *see Bailey, supra* note 203.

247. *Alice Corp. v. CLS Bank Int'l*, 573 U.S. 208 (2014); *Mayo Collaborative Servs. v. Prometheus Labs., Inc.*, 566 U.S. 66 (2012); *Bilski v. Kappos*, 561 U.S. 593 (2010); *Parker v. Flook*, 437 U.S. 584 (1978); *Gottschalk v. Benson*, 409 U.S. 63 (1972); Zivojnovic, *supra* note 51, at 811.

248. *Alice*, 573 U.S. 208; *Mayo*, 566 U.S. 66; *Bilski*, 561 U.S. 593; *Diamond v. Diehr*, 450 U.S. 175 (1981); *Parker*, 437 U.S. 584; *Gottschalk*, 409 U.S. 63; Zivojnovic, *supra* note 51, at 811.

249. *Alice*, 573 U.S. at 224.

250. *See Quinn, supra* note 159.

251. *See Gomulkiewicz, supra* note 70.

252. *See id.*

be shorter, while the protection of the underlying functionality could be more robust.²⁵³ The rights afforded to owners of software intellectual property could differ from patent and copyright and be tailored specifically to the nature of software.²⁵⁴ Additionally, a new test for infringement upon such rights could be created to cover infringement issues specific to software, such as reverse engineering.²⁵⁵ Ultimately the extent of such protection should be made by the legislature based on the guidance of experts in the field. However, the need for a specialized form of intellectual property protection specifically for software appears to be growing.

CONCLUSION

In May 2017, the Supreme Court held in *Lexmark* that when a patentee sells his or her patented product, the patentee gives up all rights associated with the product, regardless of restrictions he or she placed on the purchaser's use of the product.²⁵⁶ This decision furthers the trend of courts lessening the ability of the software industry to protect software and other computer-implemented inventions through patent law. In light of the decreased availability of patent protection for software, companies must look for a different form of intellectual property protection. The best approach would be creating a *sui generis* system of intellectual property law to specifically govern software.

253. See Samuelson, *supra* note 238, at 492-94.

254. See *id.* at 519-24.

255. See *id.* at 525-27.

256. *Impression Prods. v. Lexmark Int'l, Inc.*, 137 S. Ct. 1523, 1529 (2017).