DECONSTRUCTING WONDERLAND: MAKING SENSE OF SOFTWARE PATENTS IN A POST-*ALICE* WORLD

Joseph Allen Craig[†]

Software has long been problematic for the world of intellectual property as it partially fits into copyright, trade secret, and patent law, creating confusion about the best means of protection.[1] Computer code imperfectly fits into the copyright realm: although it receives copyright protection as a literary work, software is inherently functional, limiting the copyright protection.[2] Because there is an increasing transition from copies of software on an individual computer to cloud-based computing, where the underlying code is hidden from end users, trade secret law is also an attractive form of protection.[3] However, software also possesses the ability to be a useful, novel, and nonobvious invention, thereby fitting into the world of patent law.[4]

DOI: https://dx.doi.org/10.15779/Z38D21RJ3G

[†] J.D. Candidate, 2018, University of California, Berkeley, School of Law.

[1] *See* Gregory J. Maier, *Software Protection—Integrating Patent, Copyright and Trade Secret Law*, 69 J. PAT. & TRADEMARK OFF. SOC'Y 151, 151 (1987) ("It is the hybrid nature of software that causes its failure to fit neatly into any one existing category of intellectual property, resulting in seemingly endless confusion as to how it may best be protected."); *see also* Peter S. Menell, *The Challenges of Reforming Intellectual Property Protection for Computer Software*, 94 COLUM. L. REV. 2644, 2652–53 (1994).

[2] 17 U.S.C. § 102 (2012) (Copyright protects "original works of authorship fixed in any tangible medium of expression."); *see also* Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2316 (1994) ("While conceiving of programs as texts is not incorrect, it is seriously incomplete. A crucially important characteristic of programs is that they behave; programs exist to make computers perform tasks.").

[3] *See* Peter S. Menell, *Envisioning Copyright Law's Digital Future*, 46 N.Y.L. Sch. L. Rev. 63, 73 (2002). Prior to the 1980s, trade secret was the primary means of protecting computer software. *Id.* However, as software began to enter larger markets, companies began to worry that trade secret protection would not sufficiently protect their software products. *Id.*; *see also* Robert C. Scheinfeld & Gary M. Butter, *Using Trade Secret Law to Protect Computer Software*, 17 RUTGERS COMPUTER & TECH. L.J. 381, 418 (1991).

[4] 35 U.S.C. § 101 (2012) ("Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.").

Courts have struggled when applying traditional patent law tests for eligibility to computer software due to the lack of bright-line rules.[5] Historically, the patent protection afforded to software has ebbed and flowed, from disallowing patents for any mathematical algorithms to allowing patents for any software that creates "a useful, concrete and tangible result."[6] The 2014 Supreme Court decision in *Alice Corp. Pty. Ltd. v. CLS Bank International*[7] once again reignited the debate over the degree to which software should be patent eligible.[8] Although *Alice* did not announce a per se rule against software patents, it created a patent-eligibility test that made it difficult for software to be patented, and it called many software patents into question, prompting many decisions invalidating software patents.[9] Now, the Federal Circuit appears to be augmenting *Alice's* two-part test by using what amounts to a "technological arts" test,[10] which asks whether or not the claims are directed to a technological solution to a technological problem.

This Note begins by discussing software technology, how that technology has changed, and how the patent system has historically handled software patents. Part II summarizes the first five cases after *Alice* in which the Federal Circuit found software to be patent eligible. Finally, Part III synthesizes the previously discussed cases and addresses problems with the application of the patent-eligibility test.

[5] Andrew Beckerman-Rodau, *What Should Be Patentable?—A Proposal for Determining the Existence of Statutory Subject Matter Under 35 U.S.C. Section 101*, 13 WAKE FOREST J. BUS. & INTELL. PROP. L. 145, 147–48 (2013) (arguing that the Supreme Court has rejected bright-line tests in favor of a "more open-ended philosophical approach").

[6] *See* Brieanna Dolmage, *The Evolution of Patentable Subject Matter in the United States*, 27 WHITTER L. REV. 1023, 1034 (2006).

[7] Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347 (2014).

[8] *See e.g.*, Peter D. Junger, *You Can't Patent Software: Patenting Software Is Wrong*, 58 CASE W. RES. L. REV. 333 (2008); Robert P. Merges, *Software and Patent Scope: A Report from the Middle Innings*, 85 TEX. L. REV. 1627 (2007).

[9] *See* OIP Techs., Inc. v. Amazon.com, 788 F.3d 1359, 1364 (Fed. Cir. 2015); Internet Patents Corp. v. Active Network Inc., 790 F.3d 1343, 1348–49 (Fed. Cir. 2015); Content Extraction & Transmission LLC v. Wells Fargo Bank, Nat'l Ass'n, 776 F.3d 1343, 1349, 1351 (Fed. Cir. 2014); Ultramercial, Inc. v. Hulu, LLC, 772 F.3d 709, 717 (Fed. Cir. 2014).

[10] Austin Steelman, Note, *Curiouser and Curiouser! Why the Federal Circuit Can't Make Sense of* Alice, 98 J. PAT. & TRADEMARK OFF. SOC'Y 374, 384 (2016) ("[T]he technological arts test seeks to provide patent-eligibility for scientific innovation but not for social/commercial/economic innovation.").

I. BACKGROUND AND HISTORY OF SOFTWARE PATENTS

This Part discusses the technological aspects of software, and how software has evolved into what it is today. Additionally, it explains how patent law has attempted to keep up with the evolving software landscape.

A. SOFTWARE TECHNOLOGY

Software is a program used to direct the operation of a computer.[11] Software can take different forms, from system software[12] that runs basic computer functionality, such as the Windows operating system, to specialized application software, such as Microsoft Excel.[13] In recent years, some software applications have also been moving into the cloud, such as Google's spreadsheet program, Google Sheets.[14] Customers can access applications run on the cloud remotely over the Internet, a concept referred to as software as a service, or SaaS.[15]

In addition to different types of software, there are multiple levels of abstraction in any piece of software.[16] At its most basic concrete level, software is a set of instructions, called binary code, readable only by the computer itself.[17] However, software is typically written in a higher level language, such as Java or C++,[18] which allows programmers to write code in an English-like syntax, rather than in binary code.[19] An intermediary

[11] *Software*, DICTIONARY.COM, http://www.dictionary.com/browse/software?s=t (last visited Feb. 7, 2017) [https://perma.cc/C7MM-MRXB].

[12] *See* DAVID A. PATTERSON & JOHN L. HENNESSY, COMPUTER ORGANIZATION AND DESIGN: THE HARDWARE/SOFTWARE INTERFACE 10 (4th ed. 2009) (System software is "[s]oftware that provides services that are commonly useful, including operating systems, compilers, loaders, and assemblers.").

[13] *See Windows*, MICROSOFT, https://www.microsoft.com/en-us/windows (last visited Feb. 7, 2017) [https://perma.cc/ZRM9-W34A]; *Excel*, MICROSOFT, https://products.office.com/en-us/excel (last visited Feb. 7, 2017) [https://perma.cc/HRS5-6WZH].

[14] *Google Sheets*, GOOGLE, https://www.google.com/sheets/about/ (last visited Feb. 7, 2017) [https://perma.cc/2MHX-6E7U].

[15] *SaaS: Software as a Service*, SALESFORCE.COM, https://www.salesforce.com/saas/ (last visited Mar. 20, 2017) [https://perma.cc/F4PS-EC8U].

[16] *See* PATTERSON, *supra* note 12, at 20 ("[D]elving into the depths of hardware or software reveals more information or, conversely, lower-level details are hidden to offer a simpler model at higher levels. The use of such layers, or abstractions, is a principal technique for designing very sophisticated computer systems.").

[17] *See id.* at 11.

[18] *See id.* at 12.

[19] *See id.* at 13 ("High-level programming languages offer several important benefits. First, they allow the programmer to think in a more natural language, using English words and algebraic notation, resulting in programs that look much more like text than like tables of cryptic symbols.").

language, assembly language, falls between the binary and high-level language.[20]

B. EVOLUTION OF SOFTWARE

The world of software is rapidly changing.[21] Modern software can have several billion lines of code, creating new and unique challenges and requiring large teams of developers.[22] Software has revolutionized entire industries, such as transportation and lodging, allowing anyone with a smartphone to turn one's car into a taxi with Uber, or one's home into a hotel with AirBnB.[23] Advances in software are incredibly important to the U.S. economy and will continue to change the country.[24] Therefore, the ways in which the United States incentivizes software innovations are important not only to technology companies, but to everyone.

C. THE PURPOSE OF PATENT LAW

The Constitution established patent law to promote the advancement of science and the useful arts.[25] Patent law gives inventors limited monopolies for their publicly disclosed inventions.[26] These limited monopolies allow patent owners to exclude others from making, using, offering for sale, or selling their invention for a limited time, after which the inventions enter

[20] *See id.* at 11.

[21] *See id.* at 4 (explaining that applications that were once "economically infeasible" or "computer science fiction," such as cell phones, the human genome project, and search engines, now affect "almost every aspect of our society").

[22] *See id.* at 10 ("A typical application, such as a word processor or a large database system, may consist of millions of lines of code and rely on sophisticated software libraries that implement complex functions in support of the application."); Cade Metz, *Google Is 2 Billion Lines of Code—and It's All in One Place*, WIRED (Sept. 16, 2015, 10:00 AM), https://www.wired.com/2015/09/google-2-billion-lines-codeand-one-place/ [https://perma.cc/JF54-Q62F] (noting that Google has 25,000 developers).

[23] *See* Alamea Deedee Bitran, *The Uber Innovations that Lyfted Our Standards Out of Thin Air[Bnb], Because Now, "There's an App for That,"* 8 ELON L. REV. 503, 509, 513 (2016).

[24] *See Overview: The Economic Impact of Software*, BSA | THE SOFTWARE ALLIANCE, http://softwareimpact.bsa.org/pdf/Economic_Impact_of_Software_Overview.pdf (finding that the impact of software on the U.S. economy alone was over $1 trillion in 2014) [https://perma.cc/BVL3-85BG].

[25] U.S. CONST. art. I, § 8, cl. 8.

[26] *Id.*

the public domain.[27] These exclusive rights incentivize inventors to innovate and disclose their inventions.[28]

The Supreme Court has used § 101 as a tool to prevent patents from preempting entire fields, believing that tying up the key building blocks of invention will "impede innovation more than it would tend to promote it."[29] Accordingly, granting patents for the basic building blocks of computer code could prevent innovation, impeding the advancement of computer science.

D. History of Software Patent Eligibility

Software patent eligibility has fluctuated over the years, from software being virtually patent ineligible to all software being patent eligible.[30] This Section examines the software patent-eligibility jurisprudence.

In *Gottschalk v. Benson*, the Court held that computer algorithms were not patent eligible.[31] The claims at issue were for an algorithm that converted binary coded decimals into true binary numbers using a mathematical formula.[32] The Court found that the algorithm was merely an abstract idea and that allowing a patent on the algorithm would preempt the underlying mathematical formula.[33] The Court also introduced the

[27] 35 U.S.C. § 271(a) (2012) ("Except as otherwise provided in this title, whoever without authority makes, uses, offers to sell, or sells any patented invention, within the United States or imports into the United States any patented invention during the term of the patent therefor, infringes the patent.").

[28] *See* Bowman v. Monsanto Co., 133 S. Ct. 1761, 1768 (2013) (explaining that if an "undiluted patent monopoly" extended only for one transaction, and not for 20 years as promised by the Patent Act, a patent would plummet in value after the first sale of the first item containing the invention, resulting in "less incentive for innovation than Congress wanted"). For discussion on whether patents are properly satisfying their goal of incentivizing inventions, see Paul Belleflamme, *Patents and Incentives to Innovate: Some Theoretical and Empirical Economic Evidence,* 13 ETHICAL PERSPS.: J. OF THE EUR. ETHICS NETWORK 267, 274–78 (2006).

[29] Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347, 2354 (2014); *see also* Gottschalk v. Benson, 409 U.S. 63, 68 (1972).

[30] *See* Dolmage, *supra* note 6, at 1026 ("Although software is patentable today, it was not always so. Both the USPTO and the courts have spent decades laboring over the determination of whether software is patentable subject matter.").

[31] *Gottschalk*, 409 U.S. at 68, 71–72.

[32] *Id.* at 64. Binary coded decimals have each individual digit coded in binary, whereas true binary numbers represent the number as a whole. *Id.*

[33] *Id.* at 68. Patenting this algorithm would have had a large preemptive effect as the algorithm could be used in software for everything from "the operation of a train . . . to researching the law books for precedents . . . ." *Id.* Allowing a patent would have prevented the algorithm from being used in any other conceivable program without a license. *Id.*

"machine-or-transformation test,"[34] holding that the "[t]ransformation and reduction of an article 'to a different state or thing' is the clue to the patentability of a process claim that does not include particular machines."[35]

Next, in *Parker v. Flook* the Court held that novel algorithms added to existing processes were not patent eligible, expanding *Gottschalk*'s holding.[36] The claims at issue dealt with a method for updating an alarm limit in a catalytic conversion process.[37] The Court held that the only difference between the claims and the prior art was the specific algorithm used to update the alarm limit.[38] The Court determined that the claims were no more than a new method for calculating the alarm limit and were therefore not patent eligible.[39] Although the Court mentioned that its decision did not imply that all software would be patent ineligible, the combination of *Gottschalk* and *Flook* arguably eliminated all software patents.[40] *Gottschalk* held that algorithms were patent ineligible, and *Flook* said that novel algorithms added to existing processes were patent ineligible.[41]

In *Diamond v. Diehr*, the Court departed from its previous jurisprudence and for the first time found software patentable.[42] The claims at issue were for a process for curing synthetic rubber, which ensured that the molded articles were properly cured.[43] This process involved software containing an algorithm to dynamically update the curing time for rubber based on the rubber's temperature, which was measured periodically.[44] Importantly, the claims included steps for heating and removing the rubber in addition to the

[34] The Federal Circuit may be applying once again the machine-or-transformation test, although specifically for biological patents. *See* Joyce Li, Note, *Preemption and Diagnostics: The Federal Circuit's Misguided Return to the Machine-or-Transformation Test*, 32 BERKELEY TECH. L.J. 24–26 (2017).

[35] *Gottschalk*, 409 U.S. at 70.

[36] Parker v. Flook, 437 U.S. 584, 594–95 (1978).

[37] *Id.* at 585.

[38] *Id.* at 594.

[39] *Id.* at 594–95.

[40] Ognen Zivojnovic, Note, *Patentable Subject Matter After* Alice*—Distinguishing Narrow Software Patents from Overly Broad Business Method Patents,* 30 BERKELEY TECH. L.J. 807, 813 (2015) ("The combination of *Benson* and *Flook* virtually eliminated patent protection for software—*Benson* labeled algorithms, and thus by extension all software, as patent-ineligible abstract ideas, and *Flook* by indicating that, as long as an invention's sole point-of-novelty lay in the software, no additional limitation could be "enough" to confer patent eligibility.").

[41] Gottschalk v. Benson, 409 U.S. 63, 68, 71–72 (1972); *Flook*, 437 U.S. at 594–95.

[42] Diamond v. Diehr, 450 U.S. 175, 192–93 (1981).

[43] *Id.* at 177.

[44] *Id.*

algorithm.[45] Here, the Court found that the claims as a whole were directed to the entire process of curing the rubber, not just the algorithm involved in the software.[46] Although this case did not expressly overrule *Flook*, it can be difficult to distinguish these two cases.[47]

Software patents became easier to obtain after the Federal Circuit decision *State Street Bank & Trust Co. v. Signature Financial Group, Inc*, where the court held that an abstract idea that produces a useful and tangible result was patent eligible.[48] The claims were for a hub and spoke mutual fund investment system run on a computer,[49] which enabled individual mutual funds to pool their assets together into one investment vehicle.[50] The Federal Circuit held that the claims were patent eligible because they transformed data by using a "machine through a series of mathematical calculations."[51] This interpretation allowed people to evade the subject matter requirement by reciting traditional business methods and adding computer language to the claims.[52]

But the subject matter requirement returned in *Bilski v. Kappos*, where the Supreme Court held that a patent reciting a hedging method was ineligible because it preempted an abstract idea.[53] The Court rejected the "machine-or-transformation test" as the only test for patent eligibility, holding that it was merely one test to be considered.[54] The Court also confirmed that any "useful, concrete, and tangible result" is not automatically patentable.[55]

In *Alice Corp. Pty. Ltd. v. CLS Bank Int'l*, the Court distilled a two-part test from the 2012 product-of-nature case *Mayo Collaborative Services v. Prometheus Labs., Inc.*[56] and applied it to software.[57] The claims in *Alice*

[45] *Id*. at 177–78.

[46] *Id*. at 192.

[47] *See* Bernard Chao, *Finding the Point of Novelty in Software Patents*, 28 BERKELEY TECH. L.J. 1217, 1235 (2013).

[48] State St. Bank & Tr. Co. v. Signature Fin. Grp., 149 F.3d 1368, 1373 (Fed. Cir. 1998).

[49] *Id*. at 1370.

[50] *Id*. at 1371.

[51] *Id*. at 1373.

[52] *See* Robert A. Hulse, *Patentability of Computer Software After* State Street Bank & Trust Co. v. Signature Financial Group, Inc.*: Evisceration of the Subject Matter Requirement,* 33 U.C. DAVIS L. REV. 491, 519 (2000).

[53] Bilski v. Kappos, 561 U.S. 593, 611–12 (2010).

[54] *Id*. at 604.

[55] *Id*.

[56] *See* Mayo Collaborative Servs. v. Prometheus Labs., Inc., 132 S. Ct. 1289 (2012).

[57] Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347, 2355–60 (2014).

were directed to a form of escrow in which two parties could enter into a contract and agree to exchange money at a later time.[58] The *Alice* test has two parts.[59] First, the court must decide whether or not the claims are directed to an abstract idea.[60] Second, if the claims are directed to an abstract idea then the court must ask whether or not there is an "inventive concept" that is sufficient to make the claims patent eligible.[61] The Court found the claims were directed to the abstract idea of escrow between two parties.[62] And there was no inventive concept because the claims were nothing more than a generic computer implementation of an abstract idea.[63] Therefore the claims were patent ineligible.[64]

In the aftermath of *Alice*, many software patents were declared invalid as it was no longer possible to draft valid claims for generic processes by adding the words "on a computer."[65]

II. POST-*ALICE* FRAMEWORK FOR PATENT ELIGIBILITY

This Part discusses the first five cases since *Alice* in which the Federal Circuit held software patents to be valid. It then distills a framework for inventors hoping to draft patent-eligible claims for software-based inventions. Mainly, software claims should offer a technical solution to a technical problem.

A. CLAIMED SOLUTIONS ROOTED IN TECHNOLOGY MIGHT BE PATENT ELIGIBLE: *DDR HOLDINGS, LLC V. HOTELS.COM, L.P.*

DDR Holdings, LLC v. Hotels.com, L.P. introduced the idea that claimed solutions necessarily rooted in technology that solve problems arising in the realm of computer networks might be patent eligible.[66]

The claims in *DDR* were directed to the idea of generating a webpage on a host site with the same look and feel of the host site, but containing

[58] *Id.* at 2352.

[59] *Id.* at 2355.

[60] *Id.*

[61] *Id.*

[62] *Id.* at 2357.

[63] *Id.* at 2359–60.

[64] *Id.*

[65] *See* Internet Patents Corp. v. Active Network Inc., 790 F.3d 1343, 1348–49 (Fed. Cir. 2015); OIP Techs., Inc. v. Amazon.com, 788 F.3d 1359, 1363–64 (Fed. Cir. 2015); Content Extraction & Transmission LLC v. Wells Fargo Bank, Nat'l Ass'n, 776 F.3d 1343, 1347–48 (Fed. Cir. 2014); Ultramercial, Inc. v. Hulu, LLC, 772 F.3d 709, 714 (Fed. Cir. 2014).

[66] DDR Holdings, LLC v. Hotels.com, L.P., 773 F.3d 1245, 1257 (Fed. Cir. 2014).

content from a third-party merchant.[67] This new system allowed users to purchase items from a third-party merchant without those users navigating to a third-party website.[68] Keeping a user on a host website improved e-commerce for the host site, as it increased the chance of a user making a purchase.[69] The USPTO and the district court both found that the patents were valid.[70]

On appeal, the Federal Circuit agreed that the patents were patent eligible[71] but did not explicitly state whether the claims passed the abstract idea first step of the *Alice* test.[72] Instead, the Federal Circuit held that regardless of whether or not the claims were directed to an abstract idea, they satisfied step two because they contained an inventive concept.[73]

The court stressed that the claims were necessarily rooted in computer technology.[74] The problem that the claims addressed, keeping a user from navigating to a third-party website, was an internet-specific problem.[75] And the claims solved an internet problem by changing the way the traditional internet routine operates.[76]

The court also noted that the claims did not deal with any mathematical algorithm or fundamental business practice.[77] Further, the court emphasized that the solution did not preempt every possible solution and was therefore not overly preemptive.[78] The specificity of the solution supported the finding that it was more than the monopolization of an abstract idea.[79]

[67] *Id*. at 1248.

[68] *Id*. The prior art allowed third-party merchants to lure visitors away from the host site when the user clicked on a third-party advertisement. *Id.*

[69] *Id*. at 1257.

[70] *Id*. at 1250–51.

[71] *Id*. at 1259.

[72] *Id*. at 1257.

[73] *Id*.

[74] *Id*. ("[T]he claimed solution is necessarily rooted in computer technology in order to overcome a problem specifically arising in the realm of computer networks.").

[75] *Id*. ("Although the claims address a business challenge (retaining website visitors), it is a challenge particular to the Internet.").

[76] *Id*. Traditional internet hyperlinks would transport a visitor instantly away from a host's website after clicking on an advertisement. *Id.*

[77] *Id*. at 1257. The dissent attempted to analogize the "store within a store" concept to kiosks in a brick-and-mortar store, arguing that the idea of a store within a store has been in widespread use and is therefore neither an inventive concept nor inherently rooted in computer technology. *See id.*

[78] *Id*. at 1259.

[79] *Id*.

B. NOT ALL SOFTWARE IS ABSTRACT: *ENFISH, LLC V. MICROSOFT CORP.*

Enfish, LLC v. Microsoft Corp. held that not all software claims were inherently directed to an abstract idea.[80] The claims in Enfish's patent attempted to improve the traditional relational model database,[81] a database that uses several different tables to organize data. For example, a traditional database may use separate tables for people and companies.[82] The people table would contain the person's name, along with a link to the company table.[83] That link would refer to an entirely separate table where the company information was stored.[84]

In contrast, the claims in *Enfish* were directed to a self-referential database.[85] A self-referential database is different in that it allows for all of the database information to be incorporated into a single table.[86] A self-referential database has many advantages over a traditional relational database, including more effective storage of differing data types, ease of setup, and faster search time.[87]

At trial, the district court found that all claims were invalid as ineligible under § 101[88] because they were directed to the abstract idea of "storing, organizing, and retrieving memory in a logical table."[89] The Federal Circuit, however, pointed out that the district court oversimplified the inventive components and downplayed the benefits.[90] Significantly, the Federal Circuit determined that the patents at issue were not directed to an abstract idea under step one of the *Alice* test.[91] This showed that the first step was actually a meaningful one and that some software claims were not directed

[80] Enfish, LLC v. Microsoft Corp., 822 F.3d 1327, 1335 (Fed. Cir. 2016).

[81] *Id.* at 1330 (contrasting the patented logical model to the standard relational model).

[82] *Id.*

[83] *Id.* at 1330–31.

[84] *Id.*

[85] *Id.* at 1330, 1332.

[86] *Id.*

[87] *Id.* at 1333.

[88] *Id.*

[89] *Id.* at 1337.

[90] *Id.* at 1337–38. The district court may in fact have misunderstood the invention because the district court claimed that the inventive concept could be satisfied by adding a header row to a table, which is inaccurate. *See id.* at 1338 ("The court determined that the patents' self-referential concept could be satisfied by creating a table with a simple header row. But that is simply not the case.").

[91] *Id.* at 1339.

to an abstract idea.[92] The court also stipulated that the test was whether the character of the claims as a whole is directed to an abstract idea.[93]

The Federal Circuit reiterated that in *Alice*, the Court said that claims improving either the computer itself or a technological process might not be considered an abstract idea.[94] Here, the claims as a whole were directed to a technological improvement (a self-referential database) over a technological process (database technology).[95] The Federal Circuit held that the district court described the claims at too high a level of abstraction.[96] Any invention can to some degree be thought of as directed to an abstract idea at some level.[97] Therefore, the court must set the level of abstraction in an appropriate manner.[98]

Importantly, the Federal Circuit reiterated that the software's ability to run on a general-purpose computer does not doom the claims.[99] This is significant because ability to run on a variety of systems is often an important goal of software development.[100] The court also specified that a lack of physical components does not doom the claims either, noting that many software advancements do not require any specific hardware components.[101]

C. ABSTRACT SOFTWARE MAY CONTAIN INVENTIVE CONCEPT: *BASCOM GLOBAL INTERNET SERVICES V. AT&T MOBILITY LLC*

In *BASCOM Global Internet Services v. AT&T Mobility LLC*, the Federal Circuit found that claims that were clearly abstract were nonetheless patent eligible.[102] Here, the patents at issue were directed to filtering content on the Internet.[103]

[92] *Id.*

[93] *Id.* at 1335.

[94] *Id.*

[95] *Id.* at 1337.

[96] *Id.*

[97] *See id.* (citing Diamond v. Diehr, 450 U.S. 175, 189 n.12 (1981) (holding that if overgeneralizing claims were taken to an extreme, "all inventions [would be] unpatentable because all inventions can be reduced to underlying principles of nature, once known, make their implementation obvious.")).

[98] *See id.*

[99] *Id.* at 1338.

[100] *See id.* at 1339.

[101] *Id.*

[102] BASCOM Glob. Internet Servs. v. AT&T Mobility LLC, 827 F.3d 1341, 1352 (Fed. Cir. 2016).

[103] *Id.* at 1344.

In the prior art, internet filtering was done with software installed on a user's computer, on the local server, or at the internet service provider (ISP) level.[104] Filtering at each of these three levels presented various benefits and drawbacks. First, software on the user's computer allowed for better individual customization for the end user but the system was susceptible to modification and updating the website blocking list was difficult.[105] Second, software at the local-server level allowed companies or organizations to easily block sites from all of their users; however, this required a one-size-fits-all approach that was time consuming to maintain.[106] And third, ISPs could filter websites for all of their subscribers, but this suffered from the same problems of the one-size-fits-all approach.[107]

The claims at issue in BASCOM's patent combined the benefits of the three different methods, while mitigating their individual drawbacks.[108] The patent teaches individually customized filtering at the remote ISP server level by using the subscriber network transmission information to distinguish between requests from different users, while also allowing for a master-inclusive list along with individually customizable inclusive and exclusive lists.[109]

At trial, the district court held that BASCOM's patent was invalid as a matter of law, because it was directed to the abstract idea of filtering content under step one of *Alice*[110] and there was no inventive concept under step two of *Alice* because each individual claim limitation, taken in isolation, was a generic computer component.[111]

The Federal Circuit agreed with the district court that the claim was directed to the abstract idea of filtering content.[112] The Federal Circuit noted that the claims were, at best, ambiguously directed to an improvement in computer capabilities, although the court admitted it was a close call.[113]

Turning to step two of the *Alice* test, the Federal Circuit reversed the district court and found that there was an inventive concept.[114] It held that

[104] *Id.*

[105] *Id.*

[106] *Id.*

[107] *Id.*

[108] *Id.*

[109] *Id.* at 1344–45.

[110] *Id.* at 1346–47. The claims were compared to a librarian or parent who forbids children from reading certain books at the library. *Id.* at 1346.

[111] *Id.* at 1347.

[112] *Id.* at 1348.

[113] *Id.* at 1349.

[114] *Id.* at 1350–51.

the district court erred by looking only at the claim elements individually, instead of looking at the claim elements as an ordered combination.[115] The inventive concept was the installation of a filtering tool at a remote location with customizable filtering options for each end user.[116] The court found that BASCOM's use of network technology to associate individual accounts with the proper filtering scheme cannot be said to have been conventional or generic.[117] In other words, the patent is specifically claiming a technology-based solution to filtering internet content that overcomes some of the downsides to traditional internet filtering systems.[118]

D. Focusing on Preemption: *McRO, Inc. v. Bandai Namco Games America Inc.*

The decision in *McRO, Inc. v. Bandai Namco Games America Inc.* is important because of its focus on preemption.[119] The patents at issue in *McRO* were directed at methods for automating the lip synching and facial expressions for animated characters.[120] The two patents at issue both dealt with automating the animation process of setting keyframes, which are important visual frames in the animation.[121] This invention helped animators achieve better and faster results than was previously possible.[122]

At trial, the district court held that the claims were ineligible because they were too preemptive to satisfy § 101.[123] In the court's view, the claims attempted to patent the abstract idea of using rules to automatically set keyframes, not just one specific method for setting keyframes.[124]

The Federal Circuit determined that, similar to *Enfish*, the claims were not directed to an abstract idea under step one of the *Alice* test.[125] The Federal Circuit admonished the district court for oversimplifying the claims and overlooking the specific claim limitations.[126] Instead, the court looked at the claim limitations, as well as information from the specification, and

[115] *Id.* at 1349–50.

[116] *Id.* at 1350.

[117] *Id.*

[118] *Id.*

[119] McRO, Inc. v. Bandai Namco Games Am. Inc., 837 F.3d 1299, 1314–16 (Fed. Cir. 2016).

[120] *Id.* at 1303.

[121] *Id.* at 1305–07.

[122] *Id.* at 1306–07.

[123] *Id.* at 1309.

[124] *Id.*

[125] *Id.* at 1316.

[126] *Id.* at 1313.

determined that the rules for automating the animation were limited to rules with certain common characteristics.[127]

Significantly, this decision focused on preemption[128] because the claims were claiming a "genus" of an invention, which created a greater risk of preemption.[129] However, because there were already known alternative methods of achieving the same end result, it was not overly preemptive.[130] As the Federal Circuit made clear, the claims do not preempt "all techniques for automating 3-D animation that rely on rules."[131]

This case is also significant because the claims were not directed to a specific improvement in general computing or internet technology, but rather, the claims were "limited to a specific process."[132] As the court discussed, the patents focus on a specific means of improving the relevant technology and not merely on the end result of an abstract idea.[133]

E. ARTICULATING THE TECHNOLOGICAL ARTS TEST: *AMDOCS (ISR.) LTD. V. OPENET TELECOM, INC.*

In *Amdocs (Isr.) Ltd. v. Openet Telecom, Inc.*, the Federal Circuit found that network accounting patents were patentable using the technological arts test.[134] The four patents at issue in *Amdocs*[135] related to a system "designed to solve an accounting and billing problem faced by network service providers."[136] The patents related to improving network accounting data collection by using a distributed architecture.[137] This architecture minimized the impact on network and system resources by allowing the data to reside close to the information sources, thereby reducing network congestion while still allowing data to be centrally accessible.[138]

[127] *Id.*

[128] *Id.* at 1314–16.

[129] *Id.* at 1314.

[130] *Id.* at 1315.

[131] *Id.*

[132] *Id.* at 1316 (finding that "claim 1 is directed to a patentable, technological improvement over the existing, [sic] manual 3-D animation techniques").

[133] *Id.*

[134] Amdocs (Isr.) Ltd. v. Openet Telecom, Inc., 841 F.3d 1288, 1302–03 (Fed. Cir. 2016).

[135] U.S. Patent Nos. 7,631,065, 7,412,510, 6,947,984, and 6,836,797; *Amdocs*, 841 F.3d at 1290.

[136] *Amdocs*, 841 F.3d at 1291.

[137] *Id.*

[138] *Id.* at 1291–92.

The district court held that all four patents were ineligible under § 101.[139] The court determined under step one of the *Alice* test that the claims were directed to the "abstract idea of correlating two network accounting records to enhance the first record."[140] And under step two, the court did not find any sufficient inventive concept to confer eligibility.[141]

The Federal Circuit reversed, finding that the claims were patent eligible.[142] The court compared the claims to those in other cases, finding that they were closest to the claims in *BASCOM* and *DDR Holdings*.[143] Here, Amdocs's claims solved the problems caused by the massive data flow associated with large databases, like the claims in *DDR Holdings*, which solved the problem of the conventional internet hyperlink protocol "preventing websites from retaining visitors."[144] Moreover, the claims, when considered as an ordered combination, recited an invention that was "not merely the 'routine or conventional use' of technology."[145]

The Federal Circuit also concluded that, similar to *Enfish*, the claims posed an "unconventional technological solution (enhancing data in a distributed fashion) to a technological problem (massive record flows which previously required massive databases)."[146] Although the claims used generic computing components, the claim limitations required the generic components to operate "in an unconventional manner to achieve an improvement in computer functionality."[147] *Amdocs* therefore affirmed the requirement that claims provide a technological solution to a technical problem.[148]

III. DISCUSSION

Although there has been uncertainty since the Supreme Court went down the rabbit hole in *Alice,* the Federal Circuit has actually provided some guiding stars to go by. This Part begins by summarizing the current framework for deciding software patent eligibility established by the Federal Circuit post-*Alice*, and how the Federal Circuit is applying a "technological arts" test to the *Alice* two-step test. This Part concludes by

[139] *Id*. at 1290.

[140] *Id*. at 1299.

[141] *Id*.

[142] *Id* at 1302.

[143] *Id*. at 1301–02.

[144] *Id*. at 1302.

[145] *Id*. at 1301–02 (citing DDR Holdings, LLC v. Hotels.com, L.P., 773 F.3d 1245, 1259 (Fed. Cir. 2014)).

[146] *Id*. at 1300.

[147] *Id*. at 1300–01.

[148] *See id.*

discussing the problems arising from the Federal Circuit's use of the technological arts test.

A. SOFTWARE PATENT-EGLIBILITY FRAMEWORK ESTABLISHED BY THE FEDERAL CIRCUIT

This Section discusses how the Federal Circuit has applied the *Alice* test, how the application of the *Alice* test appears to be the technological arts test, and how similar tests have been applied internationally.

DDR, *Enfish*, and *McRO* highlight the Federal Circuit's evolving definition of an abstract idea. Step one of the *Alice* test asks whether or not the claims are directed to an abstract idea.[149] First, in *DDR Holdings, LLC v. Hotels.com, L.P.*, the court characterized the claims as "making two web pages look the same" and "making two e-commerce web pages look alike by using licensed trademarks, logos, color schemes, and layouts."[150] But it did not determine whether the claims were directed to an abstract idea.[151] Second, in *Enfish, LLC v. Microsoft Corp.*, the court held that the claims were not directed to an abstract idea because they dealt with a technical solution (a self-referential database) to a technical problem (the difficulty of having multiple data types in one table).[152] And third, in *McRO, Inc. v. Bandai Namco Games Am. Inc.* the court determined that the claims were directed to "a patentable, technological improvement over the existing, manual 3-D animation techniques."[153]

The Federal Circuit clearly held the claims to be directed to an abstract idea but still conferred patent eligibility in only one case: *BASCOM Global Internet Services v. AT&T Mobility LLC*.[154] There, the court found that the claims were directed to "a content filtering system for filtering content retrieved from an internet computer network."[155] However, under step two of the *Alice* test, the court found that the claims contained an inventive concept in that they claimed a "technology-based solution" to filtering content on the Internet that "overcomes existing problems with other Internet filtering systems."[156] Similar logic can be seen in *DDR*, where there

[149] Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347, 2355 (2014).

[150] *See DDR Holdings, LLC*, 773 F.3d at 1257.

[151] *See id.*

[152] Enfish, LLC v. Microsoft Corp., 822 F.3d 1327, 1337–39 (Fed. Cir. 2016).

[153] McRO, Inc. v. Bandai Namco Games Am. Inc., 837 F.3d 1299, 1316 (Fed. Cir. 2016).

[154] BASCOM Glob. Internet Servs. v. AT&T Mobility LLC, 827 F.3d 1341, 1352 (Fed. Cir. 2016).

[155] *Id.* at 1348.

[156] *Id.* at 1351.

was an inventive concept because the claims "specif[ied] how interactions with the Internet are manipulated to yield a desired result."[157]

In contrast, *Amdocs (Isr.) Ltd. v. Openet Telecom, Inc.* eschews the traditional two-part test, noting that "there is considerable overlap between step one and step two, and in some situations this analysis could be accomplished without going beyond step one."[158] The court then held that the claims were patent eligible because they were a technological solution to a technical problem.[159]

A common theme between all five cases in which the Federal Circuit found software to be patent eligible is that the invention was a technical solution to what could be considered a technical problem, which has been characterized as the "technological arts test."[160] The technological arts test attempts to provide patent eligibility for technological and scientific innovation, but not for "social/commercial/economic" innovation.[161] Although the test was previously rejected by the Board of Patent Appeals and Interferences,[162] it arguably was reincarnated by *Alice* and the Federal Circuit's interpretation of the *Alice* test.[163]

The technological arts test as applied by the Federal Circuit is similar to the software patent-eligibility scheme currently in place in Europe.[164] Europe allows for software patents that exhibit a technical solution to a technical problem.[165] Specifically, Europe allows for patents "[w]here said further effects have a technical character or where they cause the software to solve a technical problem."[166] Further, applying Europe's test to the search for an inventive concept seems to offer the same results as the Federal Circuit's use of the technological arts test.[167] Europe requires that the inventive concept or "something more" must be "identified in the results of running a computer program, as opposed to in the notion of the computer

[157] DDR Holdings, LLC v. Hotels.com, L.P., 773 F.3d 1245, 1258 (Fed. Cir. 2014).

[158] Amdocs (Isr.) Ltd. v. Openet Telecom, Inc., 841 F.3d 1288, 1294 (Fed. Cir. 2016).

[159] *Id.* at 1303–04.

[160] *See* Steelman, *supra* note 10, at 384.

[161] *Id.*

[162] *See Ex parte* Lundgren, No. 2003-2088, at 7–9 (B.P.A.I. Apr. 20, 2004).

[163] *See* Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347, 2351 (2014) ("[The claims] do not, for example, purport to improve the functioning of the computer itself or effect an improvement in any other technology or technical field."); *see also* Steelman, *supra* note 10, at 387.

[164] Rupert A. Knights & Craig A. Redinger, *Patent-Eligibility of Software Patents in the U.S. and Europe: A Post-*Alice *Consideration*, 8 LANDSLIDE 1, 7 (2015).

[165] *Id.* at 5

[166] *Id.*

[167] *Id.* at 7–8.

program itself. Those results must not be abstract in themselves and must go beyond the normal physical interactions between the software and the computer."[168]

B. PROBLEMS AND UNCERTAINTIES WITH THE TECHNOLOGICAL ARTS TEST

Although the technological arts test appears to be gaining prominence in Federal Circuit decisions, there is still uncertainty in its application. This Section discusses how the technological arts test currently may be misapplied, how preemption may play an additional factor, and how the Supreme Court has previously handled rules created by the Federal Circuit.

The Federal Circuit arguably misapplied the technological arts test in *Intellectual Ventures v. Symantec*, by holding that the claims were not patent eligible despite featuring a technical solution to a technical problem.[169] The claims were directed to an improvement in virus detection technology in cell phones, specifically, detecting viruses at the network level instead of at the individual device level.[170] This method prevents potentially damaging files from infecting mobile devices and does not require individual users to periodically update their phone's software.[171] However, the court determined that this invention was akin to throwing away traditional junk mail and, therefore, it was not patent eligible because it contained no inventive concepts.[172] At the very least, this case shows that there may be some difficulties in applying the technological arts test in some instances, especially in determining what is the technological solution and technological problem.

Although the technological arts test focuses on identifying a technical solution to a technical problem, preemption may still be an additional consideration, requiring claims to be sufficiently specific to confer patent eligibility.[173] In *Internet Patents Corp. v. Active Network, Inc.*, for example, the Federal Circuit held that a solution to the technical problem of data loss

[168] *Id.*

[169] Intellectual Ventures I LLC v. Symantec Corp., 838 F.3d 1307, 1321–22 (Fed. Cir. 2016).

[170] *Id.* at 1319.

[171] *Id.* at 1321.

[172] *Id.* at 1314. The dissent argued that the claims were similar to those at issue in *BASCOM Glob. Internet Servs. v. AT&T Mobility LLC*, 827 F.3d 1341, 1350 (Fed. Cir. 2016), and should therefore be patent eligible as they are a solution to a technological problem that was sufficiently concrete. *Id.* at 1321.

[173] *See* McRO, Inc. v. Bandai Namco Games Am. Inc., 837 F.3d 1299, 1314–15 (Fed. Cir. 2016).

occurring when users return to a previous internet form was patent ineligible.[174] This invention changed the way that normal internet traffic works and therefore appears to be the type of invention that would pass the technological arts test. However, the Federal Circuit characterized the claims as being directed to the abstract idea of retaining information in the navigation of online forms.[175] And because the claim limitations were merely generic data collection steps for maintaining the form's state, there was no inventive concept.[176] This appears to be an example of a patent failing to claim a technical solution in sufficiently concrete terms to limit its preemptive effect. As the court noted, the claim contained no restrictions on how the functionality was accomplished and failed to describe the mechanism for maintaining the form state, despite that being the major innovation.[177] The invention was therefore too broadly preemptive and not patent eligible because it claimed all the possible ways of implementing the solution to the form navigation problem.[178] This suggests that the court might be using the technological arts test with a preemption factor requiring a certain level of claim specificity to determine whether or not the claims are directed to an abstract idea.

IV. CONCLUSION

Despite the Federal Circuit's recent attempts to make sense of *Alice*, there is still uncertainty when it comes to what types of software can and cannot be patented. The Federal Circuit currently appears to be applying the "technological arts" test to the *Alice* two-step test. The "technological arts" test properly filters out generic business methods implemented with software, while allowing for inventions that are actually improvements to software, computers, and the internet itself. Software is extremely important to our society as a whole, and therefore the way in which software innovations are promoted is important. The Federal Circuit appears to have formulated a workable test that complies with the intent of *Alice*, but because of uncertainty associated with this approach, the Supreme Court should officially affirm the Federal Circuit's use of the technological arts test.

[174] Internet Patents Corp. v. Active Network Inc., 790 F.3d 1343, 1348 (Fed. Cir. 2015).

[175] *Id.* at 1348.

[176] *Id.* at 1349.

[177] *Id.* ("As the district court observed, claim 1 contains no restriction on how the result is accomplished. The mechanism for maintaining the state is not described, although this is stated to be the essential innovation.").

[178] *Id.*