
NOTES

EVERLASTING SOFTWARE

Software patents are some of the most commonly litigated, and preliminary evidence shows that they are particularly prone to ambush litigation.¹ The typical ambush litigation plaintiff obtains a broadly worded patent and alleges that it covers a wide range of technologies, some of which may not even have existed at the time of the patent.² The odd results of ambush litigation have led some commentators to call for the elimination of software patents.³ Their concerns are best illustrated with a famous example, the case of Freeny's patent.

In 1985, Charles Freeny, Jr., a computer scientist, obtained U.S. Patent No. 4,528,643 on a method of remote manufacturing of goods using intermodem communication over phone lines.⁴ Freeny realized that it was inefficient for stores to stock electronic merchandise that might never sell.⁵ Therefore, he disclosed and claimed the following solution: customers would use an "information manufacturing machine[]" (IMM) at a point of sale to order goods.⁶ The IMM would communicate via modem with a central machine located at a remote location and receive authorization to produce the good via an authorization code.⁷ The IMM would then record an electronic good, such as a song, on a material object, such as a cassette, that the customer could take home.⁸

¹ John R. Allison et al., *Patent Quality and Settlement Among Repeat Patent Litigants*, 99 GEO. L.J. 677, 695–96 (2011).

² See JAMES BESSEN & MICHAEL J. MEURER, PATENT FAILURE 66–67 (2008).

³ See, e.g., Pamela Samuelson, *Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 EMORY L.J. 1025, 1026, 1142–43 (1990) (arguing that there are already enough incentives to innovate in the software industry that patents are not needed); James Gleick, *Patently Absurd*, N.Y. TIMES, Mar. 12, 2000, § 6 (Magazine), at 44 (arguing that software should not be patentable because it is about "thought and abstraction" rather than "nuts[]and[]bolts," *id.* at 46).

⁴ U.S. Patent No. 4,528,643 col. 5 ll. 1–31 (filed Jan. 10, 1983); Tarek N. Fahmi & Elena B. Dreszer, *De Novo Review of Claims Construction or a Wasted Effort of the District Court: Interactive Gift Express, Inc. v. CompuServe, Inc.*, 19 SANTA CLARA COMPUTER & HIGH TECH. L.J. 315, 315 (2002).

⁵ '643 Patent col. 2 ll. 62–68, col. 3 ll. 1–3.

⁶ *Id.* col. 5 ll. 32–50.

⁷ *Id.* col. 5 ll. 35–68, col. 6 ll. 1–11, fig. 3.

⁸ *Id.* col. 4 ll. 35–59, col. 6 ll. 11–23. Freeny's claim 1, central to the ensuing dispute, recites:
1. A method for reproducing information in material objects utilizing information manufacturing machines located at point of sale locations, comprising the steps of:
providing from a source remotely located with respect to the information manufacturing machine the information to be reproduced to the information manufacturing machine, each information being uniquely identified by a catalog code;
providing a request reproduction code including a catalog code uniquely identifying the information to be reproduced to the information manufacturing machine requesting to reproduce certain information identified by the catalog code in a material object;

Freeny had every intention of commercializing the invention he had described.⁹ However, these efforts failed, and the patent eventually ended up at a nonpracticing entity (NPE) called E-Data Corp.¹⁰ In 1995, E-Data Corp. launched its first lawsuits alleging that the patent — even though filed years before the invention of the World Wide Web¹¹ — covered virtually all forms of e-commerce in digital products over the internet.¹² E-Data supported this conclusion by arguing that the claim terms should be interpreted according to their ordinary meaning. It argued that an IMM could be a personal computer; that a point of sale included a home; that an authorization code included IP addresses; and that a material object included a hard disk.¹³ If these definitions were inserted into the claim language, the patent would cover any e-commerce process involving a user buying an electronic good over the internet and receiving that electronic good on her hard drive.¹⁴

The defendant e-commerce companies argued that the claims should be limited according to the specification (the non-claim description of the invention). Their basic argument was that the specific implementation details in the specification, like components of the IMM or the type of authorization code used, limited the claims themselves.¹⁵

The district court found for the defendants on all issues,¹⁶ but the Federal Circuit vacated and remanded.¹⁷ The Federal Circuit sided with E-Data on every issue, except for the interpretation of “material object,” which it said had to be a portable object like a floppy disk and

providing an authorization code at the information manufacturing machine authorizing the reproduction of the information identified by the catalog code included in the request reproduction codes; and

receiving the request reproduction code and the authorization code at the information manufacturing machine and reproducing in a material object the information identified by the catalog code included in the request reproduction code in response to the authorization code authorizing such reproduction.

Id. col. 28 ll. 22–47.

⁹ See Fahmi & Dreszer, *supra* note 4, at 317.

¹⁰ *Id.* at 317–18. An NPE is a company that owns patents but does not produce products. It licenses the patents to other companies and may sue companies that refuse to pay for a license. Michael Risch, *Patent Troll Myths*, 42 SETON HALL L. REV. (forthcoming April 2012).

¹¹ See *Pre-W3C Web and Internet Background*, W3C, <http://www.w3.org/2005/01/timelines/timeline-2500x998.png> (last visited Feb. 25, 2012). Freeny himself has said, “I didn’t foresee the Internet.” Fahmi & Dreszer, *supra* note 4, at 318 (citing Edmund B. Burke, *For Once, Law Anticipates Technology: The E-Data Patent Saga*, EDUCOM REV., Mar.–Apr. 1997, at 6, 6).

¹² Fahmi & Dreszer, *supra* note 4, at 318.

¹³ See *Interactive Gift Express, Inc. v. CompuServe Inc.*, 47 U.S.P.Q.2d (BNA) 1797, 1801, 1804–06 (S.D.N.Y. 1998).

¹⁴ See Fahmi & Dreszer, *supra* note 4, at 320.

¹⁵ See *Interactive Gift Express*, 47 U.S.P.Q.2d (BNA) at 1804–08.

¹⁶ *Id.* at 1809–10.

¹⁷ *Interactive Gift Express, Inc. v. CompuServe Inc.*, 256 F.3d 1323, 1349 (Fed. Cir. 2001).

not an internal hard drive.¹⁸ Only this construction of “material object” prevented the Freeny patent from covering e-commerce in electronic goods as a whole.¹⁹

What explains cases like Freeny’s? Despite the chorus of voices calling for the reach of software patents to be pulled back, commentators have failed to provide a convincing explanation of why software patents are so prone to ambush litigation. This Note argues that software patents are different from patents on physical objects because of the inherently functional and mathematical nature of software technology. Software operates in an (almost) entirely understood world of pure mathematics, rather than in the real world, which is governed by messy physical rules. As a result, a patent to a physical apparatus can usually be designed around by creating a different apparatus that accomplishes the same *function* but uses different *physical characteristics*. However, software does not have physical characteristics and is defined — with the partial exception of software patents covering scientifically advanced technologies — by function itself. Thus, software patents are harder to design around and tend to survive a long time. These properties of breadth and non-obsolescence also mean that software patents can end up covering after-arising technologies that their inventors did not envision — leading to ambush litigation.

In the remainder of this Note, Part I describes the difference between software and physical object patents, explains that this difference arises from the mathematical and functional nature of software, and considers the intermediary case of scientifically advanced software patents; Part II discusses explanations other scholars have provided to distinguish software patents and physical object patents; and Part III addresses implications of this theory.

I. HOW SOFTWARE IS DIFFERENT

Software patents are inherently different from other patents. As section A describes, patents on regular physical objects like apparatuses and compounds inevitably involve physical characteristics, whereas software patents are claimed conceptually, without recourse to secondary characteristics. Section B explains this phenomenon: a physical object invention is highly contingent on particular physical characteristics, while software operates in a world of well-understood mathematical rules where an invention is simply a function. Section C describes how scientifically advanced software patents exist in a middle

¹⁸ See *id.* at 1335, 1338, 1340, 1342, 1344.

¹⁹ See *E-Data Corp. v. Corbis Corp.*, No. C04-1733L, 2005 WL 1838614, at *3 (W.D. Wash. Aug. 1, 2005).

ground between run-of-the-mill software patents and physical object patents, and thus are rightly treated as a combination of the two.²⁰

A. Differences in Patents

1. *Physical Object Patents.* — Patents on physical objects such as apparatuses and compounds have been recognized since the Founding.²¹ The scope of physical object patents is relatively narrow in the sense that it is often easy to “design around” such patents. Physical objects have physical characteristics, including size, structure, and (for objects with multiple components) specific ranges of motion and frictions between moving parts. Such physical characteristics are typically recited in the claims to a physical object.

For example, a popular patent law textbook uses U.S. Patent No. 5,205,473 to a “Recyclable Corrugated Beverage Container and Holder” as its canonical example of a patent.²² The text of claim 1 is:

1. A recyclable, insulating beverage container holder, comprising a corrugated tubular member comprising cellulosic material and at least a first opening therein for receiving and retaining a beverage container, said corrugated tubular member comprising fluting means for containing insulating air; said fluting means comprising fluting adhesively attached to a liner with a recyclable adhesive.²³

The claim thus recites several physical characteristics, including the physical structure of three different components (the corrugated tubular member, fluting, and liner), how some of the components are attached (adhesively), and what some components are made of (cellulosic material).

Physical characteristics in turn supply the main source of inefficiencies that later inventors can improve. By making an improvement to a physical characteristic, a later inventor can develop a second generation product that designs around the original invention.²⁴ The second gen-

²⁰ At the outset, a brief primer on the scope of patent protection may be helpful. A patent’s claims define its coverage. A later product that falls within the language of the claims is said to be infringing. However, the patentee cannot simply claim whatever she wants; rather, she can claim only that for which she has provided an adequate written description that shows “possession” of the invention and enables another to practice it. This description, called the disclosure, is provided in the other main part of the patent, the specification. In addition, the patentee cannot claim what someone else has already invented or what would be obvious. The body of preexisting work to which reference is made in testing for novelty and nonobviousness is known as the prior art.

²¹ Peter S. Menell, *Forty Years of Wondering in the Wilderness and No Closer to the Promised Land: Bilski’s Superficial Textualism and the Missed Opportunity to Return Patent Law to Its Technology Mooring*, 63 STAN. L. REV. 1289, 1292–95 (2011).

²² ROBERT PATRICK MERGES & JOHN FITZGERALD DUFFY, PATENT LAW AND POLICY 14–23 (4th ed. 2007).

²³ U.S. Patent No. 5,205,473 col. 4 ll. 63–68, col. 5 ll. 1–2 (filed Mar. 19, 1992).

²⁴ One study has found that the ease of designing around a patent is the most significant reason for companies not to pursue a patent. Wesley M. Cohen et al., *Protecting Their Intellectual*

eration product can perform the same function — ideally even more effectively — without literally infringing the original patent claims.

For example, in *Cordis Corp. v. Boston Scientific Corp.*,²⁵ the patent at issue covered a coronary stent for the treatment of coronary artery disease, which is caused by a narrowing of the arteries.²⁶ The stent is mounted on an angioplasty balloon and inserted into an artery.²⁷ After being moved into the narrow portion of the artery, the balloon is expanded, causing the stent to expand as well.²⁸ The balloon is then deflated and withdrawn, but the stent is left in place to hold the artery open.²⁹

The alleged infringer's stent worked in exactly the same way, but no literal infringement was found because the physical design of the stent was subtly different. The patent claimed a metal stent having a surface divided into cells, each of which had an undulating curve with a crest and a trough.³⁰ However, the alleged infringer's stent had cells containing only a U-shaped curve.³¹ It thus had a trough, but no crest.³² In other words, the alleged infringer had found a different physical design that could perform the same function in a better, or at least alternative, way.³³

Assets: Appropriability Conditions and Why U.S. Manufacturing Firms Patent (or Not) 14–15 (Nat'l Bureau of Econ. Research, Working Paper No. 7552, 2000).

²⁵ 658 F.3d 1347 (Fed. Cir. 2011).

²⁶ *Id.* at 1350–51 (citing *Cordis Corp. v. Medtronic AVE, Inc.*, 339 F.3d 1352, 1354–55 (Fed. Cir. 2003)).

²⁷ *Id.* (citing *Medtronic*, 339 F.3d at 1355).

²⁸ *Id.* (citing *Medtronic*, 339 F.3d at 1354–55).

²⁹ *Id.* (citing *Medtronic*, 339 F.3d at 1355).

³⁰ At issue was claim 25, which depends on the independent claim 22. The two claims together read:

22. A pre-deployment balloon expandable stent structure adapted for percutaneous delivery to the curved coronary arteries, the stent structure being generally in the form of a thin-walled metal tube having a longitudinal axis, the stent structure having a multiplicity of closed perimeter cells, each cell having one or more undulating sections, each undulating section having a generally curved shape and having a first end point and a second end point wherein a line drawn from the first end point to the second end point is generally parallel to the stent's longitudinal axis.

25. The stent of claim 22 wherein the undulating section of each closed perimeter cell comprises a "U" shaped curve.

Id. at 1355 (emphases omitted) (quoting U.S. Patent No. 5,879,370 col. 6 ll. 17–26, 35–36 (filed May 28, 1997)).

³¹ *Id.* at 1357–58.

³² *Id.*

³³ This concept of designing around similarly applies in the biopharmaceutical context because a patent typically covers a specific composition, which can be designed around if a competitor finds a different composition that performs the same function. An example is *Duramed Pharms., Inc. v. Paddock Labs., Inc.*, 644 F.3d 1376 (Fed. Cir. 2011), which involved a patent to a "conjugated estrogen pharmaceutical composition[] for use in hormone replacement therapies." *Id.* at 1378. The patent claimed a conjugated estrogen composition coated with a moisture barrier coating (MBC) comprising ethylcellulose. *Id.* The MBC was needed to prevent breakdown of the conjugated estrogen during storage. *Id.* The alleged infringer produced a conjugated estrogen

2. *Software Patents.* — In contrast, software patents are claimed at a higher conceptual level than are physical object patents. Typically, any object recited in a software claim is named with a conceptual term rather than a term that would identify a specific object in order to ensure non-obsolescence across technological generations.³⁴ As an example, consider Amazon’s One Click patent. Claim 1 of the patent recites:

1. A method of placing an order for an item comprising:
 - under control of a client system,
 - displaying information identifying the item purchasable through a shopping cart model; and
 - in response to only a single action being performed, sending a request to order the item . . . to a server system;
 - under control of a single-action ordering component of the server system,
 - receiving the request;
 - retrieving additional information previously stored for the purchaser . . .
 - generating an order to purchase the requested item for the purchaser . . . using the retrieved additional information; and
 - fulfilling the generated order . . .
- whereby the item is ordered without using the shopping cart model.³⁵

Here, the major objects are the “client system,” “information,” “item purchasable through a shopping cart model,” “request,” “server system,” “single-action ordering component,” and so on. This conceptual terminology keeps the invention at a high level of abstraction. An “item purchasable through a shopping cart model” comprehensively includes many different types of things people could purchase, such as books, movies, or MP3s. Similarly, the patent uses conceptual language to describe the steps of the method. Major functional terms include “displaying,” “sending,” “receiving,” “retrieving,” “generating,” and “fulfilling.” These functional terms are conceptual and could be instantiated by many different kinds of real-world actions.

Because no details of the physical apparatus that performs the method need be recited, software claims will not become obsolete as a result of changes to physical technology. The One Click patent does not say anything about the kinds computers used, just that there is a “client system” and a “server system.” Software claims are completely agnostic about computing platform, from the Pentium II of 1997 to the latest 4.0 GHz processor. Just about all software methods could be performed on the first electronic computer, the ENIAC of 1946.³⁶ Un-

compound coated with an MBC made of polyvinyl alcohol. *Id.* Because the material used in the MBC was not the claimed ethylcellulose, the court found no infringement. *See id.* at 1379–82.

³⁴ *See* BESSEN & MEURER, *supra* note 2, at 67, 200.

³⁵ Ex Parte Reexamination Certificate, U.S. Patent No. 5,960,411 col. 1 ll. 23–30, col. 2 ll. 1–15 (issued July 13, 2010).

³⁶ *See infra* pp.1465–66 (describing how programs can be ported from one platform to another). However, the ENIAC would not be able to perform methods requiring a network.

doubtedly, any imaginable computer, even one thirty years in the future and based on a thousand-core processor with an optical bus, will still perform the software methods that are patented today.³⁷

The method by which this high level conceptual claiming is accomplished is the technique of *functional claiming*, which is defining elements of an invention by what they do rather than what they are.³⁸ Method claims involve functional claiming because the invention itself is a function (that is, a series of steps). Beyond method claims, courts have also accepted functional claims to computer systems and computer-readable media.³⁹ Through functional claiming, patentees need state fewer of the specifics about their inventions.

Given that the physical apparatuses and objects in a software claim are described conceptually, the only real limit — and the novelty — in the claim comes from the recitation of a method, namely a sequence of steps. However, methods obsolesce at a much slower rate than physical apparatuses.⁴⁰ A method claimed today could end up being the same method used 50, 100, or 1000 years from now.⁴¹

³⁷ That is, any traditional computer. Quantum computers, which are relegated to special purpose tasks, perform methods using different algorithms.

³⁸ *In re Swinehart*, 439 F.2d 210, 212 (C.C.P.A. 1971). Functional claiming does not require using means-plus-function claiming under 35 U.S.C. § 112, para. 6 (2006). In means-plus-function claiming, the inventor claims “means for performing function X” rather than reciting a specific structure. See U.S. PATENT & TRADEMARK OFFICE, MANUAL OF PATENT EXAMINING PROCEDURE § 2181, at 2100-234 to -237 (8th ed. Rev. 8, July 2010) [hereinafter MPEP], available at <http://www.uspto.gov/web/offices/pac/mpep/index.htm>. The inventor claims all structures equivalent to those disclosed in the specification for performing a certain function. *Id.* However, means-plus-function claiming often backfires on the patentee. Patents are often found invalid because they do not clearly disclose the structure that performs the function claimed using 35 U.S.C. § 112, para. 6. See Christa J. Laser, *A Definite Claim on Claim Indefiniteness: An Empirical Study of Definiteness Cases of the Past Decade with a Focus on the Federal Circuit and the Insolubly Ambiguous Standard*, 10 CHI.-KENT J. INTELL. PROP. 25, 37 tbls.6 & 7 (2010). In software, it is especially common for courts to find that insufficient structure is disclosed. See James Farrand et al., “Reform” Arrives in Patent Enforcement: The Big Picture, 51 IDEA 357, 418-20 (2011). What is or is not sufficient structure is unclear. Paul R. Kitch, *Step-Plus-Function: Just What Have We Stepped Into?*, 7 J. MARSHALL REV. INTELL. PROP. L. 117, 127 n.64 (2007). Therefore, patent attorneys often avoid means-plus-function claiming. See *id.* Plain vanilla functional claiming is seen as more effective.

³⁹ The typical software patent will first recite a method comprising performing steps X, Y, and Z on a computer. See, e.g., U.S. Patent No. 6,285,999 col. 8 ll. 54-68 (filed Jan. 9, 1998). Second, it will recite a computer system “configured to” perform steps X, Y, and Z. See, e.g., U.S. Patent Pub. 2009/0077056 claim 17 (filed Sept. 17, 2007). Third, it will recite a computer readable medium containing instructions that perform steps X, Y, and Z. See, e.g., ’999 Patent col. 10 ll. 38-54.

⁴⁰ Any invention can be viewed at two levels: (1) what it does and (2) how it does it. The first aspect is the method and the second aspect is the implementation details, such as what physical apparatus is used or what objects are operated upon. Therefore, abstracting away implementation details necessarily implies there are fewer ways for an invention to obsolesce. There would have to be a completely new way of doing things rather than just a change in implementation.

⁴¹ For instance, Bernard Bilski tried to patent the method of hedging risk, which is unlikely to become obsolete any time soon. See *Bilski v. Kappos*, 130 S. Ct. 3218, 3231 (2010).

B. Differences in Technologies

One may question why inventors of physical objects include physical characteristics in their claims at all. Could not all inventors make claims at a conceptual level, as in software patents, rather than defining aspects of the invention so specifically? For instance, why did the inventors in *Cordis Corp.* specify that the stent had to have undulating curves?

The key difference between object claims and software claims is that objects operate in the real world where human understanding of physics is incomplete, whereas software operates in an (almost) entirely understood world of pure mathematics. If someone claims to be able to perform a function in the real world, she must prove it by giving physical details and schematics. On the other hand, merely describing a computerized method enables a programmer of average skill to implement that computerized method on a real computer.⁴² Function itself defines software. Once the function is defined, implementation of the functionality on a computer just requires paying for programmer time.⁴³

1. *Physical Object Patents.* — Humans have never completely understood the physical world and likely never will.⁴⁴ The inventor of a new physical invention, such as a stent, must prove that she can actually implement the claimed functionality in the real world.⁴⁵ In the case of the stent, offering this proof would involve the nontrivial task of disclosing a design for a stent that could fit over a balloon, be inserted into a person's arteries, be pushed into place, expand to the right size, and stay in place over time. Clearly, the material that the

⁴² See *Fonar Corp. v. Gen. Elec. Co.*, 107 F.3d 1543, 1549 (Fed. Cir. 1997) (“[N]ormally, writing code for . . . software is within the skill of the art, not requiring undue experimentation, once its functions have been disclosed.”); Supplementary Examination Guidelines for Determining Compliance with 35 U.S.C. 112 and for Treatment of Related Issues in Patent Applications, 76 Fed. Reg. 7162, 7171 (Feb. 9, 2011) (“Computer-implemented inventions are often disclosed and claimed in terms of their functionality. This is because writing computer programming code for software to perform specific functions is normally within the skill of the art once those functions have been adequately disclosed.”); Dan L. Burk & Mark A. Lemley, *Is Patent Law Technology-Specific?*, 17 BERKELEY TECH. L.J. 1155, 1162–64 (2002).

⁴³ For example, many web portals provide access to freelance programmers who will implement any specified functionality for a small fee. See, e.g., ELANCE, <https://www.elance.com> (last visited Feb. 25, 2012).

⁴⁴ Physics lacks a single unifying theory and indeed one may never be found. STEPHEN HAWKING & LEONARD MLODINOW, *THE GRAND DESIGN* 7–8 (2010). Even if all physical laws were known, it would be impractical to solve equations for the behavior of every atom in order to predict the behavior of a macro-level object. See *id.* at 32–33; see also MORTON TAVEL, *CONTEMPORARY PHYSICS AND THE LIMITS OF KNOWLEDGE* 213–19 (2002) (describing “chaos,” a type of behavior that is very difficult to predict despite being governed by deterministic rules).

⁴⁵ See 35 U.S.C. § 112, para. 1 (2006).

stent is made of, the structural relationship between stent components, and possibly the production process would need to be disclosed. All of these physical characteristics create openings for potential design-arounds. Changing any of these physical characteristics could have effects on the amount of time needed to insert the stent, the ability of the stent to stay in place, the nontoxicity of the stent to the human body, and so forth. Thus, there are potential ways for later inventors to improve on this stent by changing the physical characteristics. Over the course of later product generations, the function of the invention does not change; rather, later inventive steps involve finding new ways to interact with the physics of the real world in order to accomplish the function in a more effective way.

Unlike inventors of software, the inventor of a new apparatus does not obtain rights to the entire genus of objects that accomplish the same function. An inventor's description of one object for accomplishing a function in the real world does not enable or describe all objects that accomplish the function.⁴⁶ Generally, human beings do not know enough about the physical world to describe all possible physical instantiations of a device that would perform a specific function. Therefore, the initial inventor of the coronary stent cannot lay claim to all possible coronary stents of all possible shapes, sizes, and structures.⁴⁷

*General Electric Co. v. Wabash Appliance Corp.*⁴⁸ demonstrates how attempts to claim all physical objects that perform a given function are likely to be invalid.⁴⁹ In that case, the inventor tried to define his apparatus by what it did rather than by its structure. At issue was claim 25, which recited:

25. A filament for electric incandescent lamps or other devices, composed substantially of tungsten and made up mainly of a number of comparatively large grains of such size and contour as to prevent substantial sagging and offsetting during a normal or commercially useful life for such a lamp or other device.⁵⁰

The Supreme Court noted that the inventor had described the grains of the filament entirely by their function, but without saying

⁴⁶ *In re Hyatt*, 708 F.2d 712, 714-15 (Fed. Cir. 1983). A claim's scope must be commensurate with the extent of the enabling disclosure, *id.*, and the inventor must provide a written description sufficient to show "possession" of the invention, *Ariad Pharm., Inc. v. Eli Lilly & Co.*, 598 F.3d 1336, 1351-52 (Fed. Cir. 2010).

⁴⁷ The first patent to a coronary stent appears to be U.S. Patent No. 6,974,475 to Dr. Henry Wall. Ron Winslow, *Will Stent Makers Fight Dentist's Patent Tooth and Nail?*, WALL STREET JOURNAL, Jan. 26, 2006, at B1. The patent claims are quite specific in the physical design of the stent that allows it to expand and stay in place as necessary. See U.S. Patent No. 6,974,475 col. 5 ll. 38-60 (filed Dec. 8, 1987).

⁴⁸ 304 U.S. 364 (1938).

⁴⁹ *Id.* at 369-72.

⁵⁰ *Id.* at 368 (quoting U.S. Patent No. 1,410,499 (filed Feb. 20, 1917)).

anything about their structure, except that they were “comparatively large.”⁵¹ The Court found the claim invalid for being too broad — the inventor had not written enough in the specification to show that he actually knew how to make all filaments that would accomplish this functionality.⁵² Thus, the inventor failed to get a patent on all physical objects performing a function because they were beyond his ability to describe.

In another case, *In re Ludtke*,⁵³ the inventor claimed a parachute that opened sequentially so as to reduce the opening shock force.⁵⁴

Claim 1 recited:

1. A parachute canopy comprising —
a plurality of circumferentially complete panels of successively larger circumferences; and
a plurality of radially extending tie lines interconnecting said panels in a radially spaced relationship said plurality of tie lines providing a radial separation between each of said panels upon deployment creating a region of high porosity between each of said panels such that the critical velocity of each successively larger panel will be less than the critical velocity of the previous panel, whereby said parachute will sequentially open and thus gradually deaccelerate.⁵⁵

The Court of Customs and Patent Appeals held that the portion of the claim relating to the sequential opening had no patentable significance because it only stated a desired result without stating how the result would be achieved.⁵⁶ Specifically, the inventor, William Ludtke, did not describe what structural relationship would cause the sequential opening of the panels.⁵⁷ Ludtke failed to show that he or anyone else knew how to make the parachute operate as claimed, much less that he knew all the ways to do so.⁵⁸ The court found the remaining patentably significant portions of the claim invalid by obviousness.⁵⁹

⁵¹ *Id.* at 369–70.

⁵² *Id.* at 371, 373–75.

⁵³ 441 F.2d 660 (C.C.P.A. 1971).

⁵⁴ *Id.* at 660.

⁵⁵ *Id.* at 661 (quoting U.S. Patent Application Serial No. 550,711 (filed May 17, 1966)).

⁵⁶ *Id.* at 661, 664.

⁵⁷ *Id.*

⁵⁸ *See id.*

⁵⁹ *Id.* Similarly, method claims to all possible physical ways of doing something in the real world are also likely to be denied. In a famous case, Samuel Morse made a broad method claim to telegraphy and was denied. He claimed “the use of the motive power of the electric or galvanic current . . . for marking or printing intelligible characters, letters, or signs, at any distances” by any means at all. *O’Reilly v. Morse*, 56 U.S. (15 How.) 62, 86 (1854). In his claim he stated: “I do not propose to limit myself to the specific machinery, or parts of machinery, described in the foregoing specifications and claims.” *Id.* The Supreme Court found this claim invalid because Morse was claiming beyond what he had disclosed. *Id.* at 112–13. Morse had not disclosed *all ways* of using the motive power of electric or galvanic current to transmit a message. *Id.*

Nonetheless, it is not the case that physical objects can never be claimed functionally. Functional claiming of physical objects is well accepted.⁶⁰ The crucial point is that the inventor can only functionally claim what she actually enabled and described.⁶¹ For physical objects, the extent of possible functional claiming is limited because it is hard for inventors to describe all possible physical objects to do something or all possible physical ways of doing something.

Furthermore, once any inventor patents or describes a single physical object that performs a function, no later inventor can claim a patent on the whole category of devices that perform the function: the existence of any single object fitting into a category makes the category no longer “new” enough to be patented.⁶² Thus, inventors of second-generation products *must* recite physical characteristics in patent claims to ensure that such claims do not cover preexisting work, such as prior patents, publications, or first-generation products.⁶³

Therefore, the only way an inventor could obtain a patent on a physical object that would cover all other objects performing the same function would be to (1) disclose a physical formula for making all objects performing the function, and (2) be the first to come up with any object performing the function. The second requirement pushes up the time at which the inventor must be able to describe all objects performing the function, which makes it even less likely that she will be able to do so. Although it is conceivable that an inventor might be able to describe all possible physical objects for performing a function in a very mature technology area, it would likely be hard for an inventor to do so for a new and immature technology that no one else has developed yet.⁶⁴ The inventor may well know only one object that performs the function, her own, thereby making generalization difficult. Therefore, it is hard for an inventor to claim all ways of implementing a given functionality in the real world.⁶⁵

2. *Software Patents.* — Unlike physical object claims, software claims can cover many or all possible ways of instantiating functionality because software is defined by function.⁶⁶ Once the function is dis-

⁶⁰ See *In re Swinehart*, 439 F.2d 210, 212–13 (C.C.P.A. 1971).

⁶¹ 35 U.S.C. § 112, para. 1 (2006).

⁶² See *In re Guess*, 347 F. App'x 558, 560 (Fed. Cir. 2009); MPEP, *supra* note 38, § 2131.02.

⁶³ The basic rule is that an inventor wants the claims to cover later products, so she can claim infringement, but not to cover any earlier products, because then her patent would be invalid for not being “new.”

⁶⁴ Cf. *O'Reilly*, 56 U.S. (15 How.) at 113 (describing how Morse had not described all possible ways of using electromagnetism to transmit a message because new discoveries in electromagnetism were still possible).

⁶⁵ Thus, this Note's conclusion is in contrast to James Bessen and Professor Michael Meurer's assertion that any kind of technology can be claimed abstractly. BESSEN & MEURER, *supra* note 2, at 66–67, 213–14.

⁶⁶ One might think of software being defined as “software that does *X*.”

closed, many ways of implementing the software are enabled and described.⁶⁷ It is the nature of software itself, not the mere choice of patent attorneys to claim software in a functional way, that causes software claims to be broad.⁶⁸

Unlike the physical characteristics of apparatuses, the secondary, nonfunctional aspects of software are unimportant to the patentable invention. Software's secondary characteristics — the nonfunctional characteristics that make it work — are qualities such as what programming language the software is written in, how much memory it occupies, and what platform and operating system the software runs on.

An average programmer can implement many versions of software that have the same functionality and yet run the entire gamut of secondary characteristics. Translating software from one programming language to another is a mostly mechanical exercise that the average programmer performs with ease.⁶⁹ Programming languages are almost completely equivalent in the types of software they can express.⁷⁰ Furthermore, any operating system and platform can run nearly any program,⁷¹ because all computer programs can, roughly speaking, be performed using the basic functions of load, store, add, subtract, conditional branching, and so on, and nearly all computers implement these basic functions.⁷² Therefore, computations performed on a given processor can be performed on any other processor, assuming that the processors have access to enough memory.⁷³ Patentees already take advantage of this equivalence across processors by making sure to claim their invention as “computer implemented” with a broad definition of “computer” that includes “personal computers, desktop computers, laptop computers, message processors, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, mobile telephones, PDAs, pagers, and the like.”⁷⁴

By contrast, it would make no sense for software patentees to specify secondary characteristics like a programming language, operating

⁶⁷ See sources cited *supra* note 42.

⁶⁸ Bessen and Meurer, on the other hand, believe that it is patent attorneys' decision to claim inventions abstractly, perhaps using functional language, that leads to patent breadth. BESSEN & MEURER, *supra* note 2, at 213.

⁶⁹ See, e.g., JEAN E. SAMMET, PROGRAMMING LANGUAGES 41–43 (1969).

⁷⁰ *Id.*

⁷¹ At worst, one can almost always emulate one platform on another. *Id.* at 41.

⁷² See, e.g., DONALD E. KNUTH, THE ART OF COMPUTER PROGRAMMING xi, 120–52 (2d ed. 1973).

⁷³ The processor must have enough memory to store the results of all its computations.

⁷⁴ U.S. Patent No. 7,778,987 col. 6 ll. 1–6 (filed Oct. 6, 2006).

system, or platform in their patents.⁷⁵ These have nothing to do with the invention.⁷⁶

The previously described Amazon One Click patent provides an illustration.⁷⁷ The main elements of the claim are steps like “displaying information,” “sending a request,” “receiving the request,” “retrieving additional information,” “generating an order,” “fulfilling the generated order,” and so on. Each individual step is something that people already know how to perform with computers in a wide variety of ways. For example, “displaying information” could be performed with different graphics protocols like OpenGL or DirectX, with different operating system APIs, on different monitors like LCDs or CRTs, and so on.⁷⁸ Similarly, “sending a request” could be performed over different kinds of networks such as a LAN, WAN, or the internet; over different low-level protocols like TCP/IP or UDP; over different application-layer protocols like HTTP, FTP, or Telnet; and so on.⁷⁹ Yet, each of these implementation possibilities is known by a person having ordinary skill in the art (PHOSITA) and is therefore enabled by disclosure of the function itself.⁸⁰ Limiting the claim to cover only certain implementations of the method, such as requiring that “sending a request” occur over the internet in a web browser using TCP/IP, would not make sense because such implementation details would be arbitrary.

⁷⁵ Note that software speed is not a secondary characteristic. Speed is not a design choice made in creating software. It is an end result of all the design choices.

⁷⁶ Some might suggest that software patents be narrowed by disclosure and limitation to a specific software architecture. See Burk & Lemley, *supra* note 42, at 1166 n.46. However, this suggestion is misguided because software architecture, like other secondary characteristics of software, is completely fungible. Although there are conventional software architectures to use in creating certain types of software, there is nothing to prevent a programmer from translating a program written with a specified software architecture into an arbitrarily different software architecture. See LEN BASS ET AL., *SOFTWARE ARCHITECTURE IN PRACTICE* 5 (2d ed. 2003) (discussing how two different software architects, when given the same requirements specification, would likely create different software architectures). One could write any computer program with almost no software architecture at all, just one monolithic function. See *id.* at 22 (“In the most trivial case, a system is itself a single element — uninteresting and probably nonuseful but an architecture nevertheless.”). Thus, the suggestion to limit software patents to specific software architectures is no better than limiting software by programming language, operating system, or platform.

This Note’s conclusions in this regard contrast with those of Professors Dan L. Burk and Mark A. Lemley, who believe that such secondary characteristics should be required in the specification and possibly the claims of software patents. Burk & Lemley, *supra* note 42, at 1163–67, 1191–92.

⁷⁷ See *supra* section I.A.2.

⁷⁸ See PETER SHIRLEY & STEVE MARSCHNER, *FUNDAMENTALS OF COMPUTER GRAPHICS* 4 (3d ed. 2009).

⁷⁹ See *Ultramercial, LLC v. Hulu, LLC*, 657 F.3d 1323, 1329 (Fed. Cir. 2011).

⁸⁰ See *id.* Several patent law doctrines, including enablement, use a standard based on what a person having ordinary skill in the art would know.

After-arising means of accomplishing the steps are also covered under current law.⁸¹ The result is sensible. Otherwise, a future programmer could avoid infringement by implementing the method using C+++ rather than C++. The critical point is that it does not matter how “sending a request” is achieved because that detail is irrelevant to the patentable invention.⁸² In software, unlike with physical apparatuses, the patentable invention is new functionality, not performing old functionality in a more effective way by harnessing physical rules.

As a further example, U.S. Patent No. 6,324,538 is a software patent that was recently asserted by an NPE against MySpace.⁸³ Claim 1 recites:

1. A method of publishing information on a computer network comprising the steps of:
creating a database entry containing information received from a user of the computer network, wherein the information includes data representing text, a universal resource locator, an image, and a user-selected category;
generating a transaction ID corresponding to the database entry;
password protecting the entries;
displaying the entries in accordance with the user-selected category;
presenting the information to a user in hyper text markup language in response to a user's request.⁸⁴

As with the One Click patent, the steps of the claim are all things that people know how to do in a variety of ways: “creating a database entry,” “generating a transaction ID,” “password protecting the entries,” “displaying the entries,” and “presenting the information.” The result is that the patentee is not limited to a specific implementation but instead lays claim to a broad way of doing things. The patent would broadly cover online systems that “enable user-generated and user-controlled content to be published and password-protected on the internet.”⁸⁵

C. Scientifically Advanced Software Patents

The treatment of scientifically advanced software patents bolsters this Note's thesis that software patents are broader because software is itself inherently functional. Where software operates on the edge of human knowledge, then, this theory would accordingly hypothesize

⁸¹ *Innogenetics, N.V. v. Abbott Labs.*, 512 F.3d 1363, 1371–72 (Fed. Cir. 2008).

⁸² See *Ultramercial*, 657 F.3d at 1329.

⁸³ *MySpace, Inc. v. GraphOn Corp.*, 756 F. Supp. 2d 1218, 1223 (N.D. Cal. 2010).

⁸⁴ U.S. Patent No. 6,324,538 col. 12 ll. 45–59 (filed July 7, 1998).

⁸⁵ *MySpace*, 756 F. Supp. 2d at 1222. Patents with wide coverage, however, are vulnerable because only one prior art species is needed for invalidation. Such invalidation occurred in *MySpace*: the patent was completely invalidated by a prior art system that was in use over eighteen months before the patent was filed. *Id.* at 1224, 1236–43.

that patent claims for such software would be less broad than for ordinary “business method”-type software. And indeed, this is the observed reality: patents on these inventions have to recite *how* they achieve their function, and not simply function alone, and thus fall somewhere in the middle ground between physical object patents and software patents.

For example, consider Google’s patent on PageRank, which is the algorithm that determines search rankings in Google.⁸⁶ Larry Page and Sergey Brin collaborated in developing the algorithm while they were Ph.D. students at Stanford.⁸⁷ Page and Brin realized that other search engines of the time relied almost entirely on the content of a web page to rank it.⁸⁸ A search for “cat” would rank pages essentially according to how many times the word “cat” appeared in the page.⁸⁹ This method led to high rankings for low-quality pages because webmasters could manipulate their ranking by using many popular keywords repeatedly.⁹⁰ The PageRank algorithm ranks page quality by determining how many other web pages link to that page.⁹¹ A page with many in-links is more likely to be a high-quality site, like NYTimes.com, than a fly-by-night blog.⁹² The PageRank patent recites the following claim:

1. A computer implemented method . . . comprising:
obtaining a plurality of documents, at least some of the documents being linked documents, at least some of the documents being linking documents, . . . each of the linked documents being pointed to by a link in one or more of the linking documents;
assigning a score to each of the linked documents based on scores of the one or more linking documents and
processing the linked documents according to their scores.⁹³

The plain text of the PageRank claims seems to cover all ways of ranking web pages based on incoming links to a page,⁹⁴ but the specification makes clear that the patent covers only a single algorithm for doing so. The specification recites that the “invention” requires using a particular formula for a page’s rank, which can be calculated by a

⁸⁶ However, now many additional heuristics are applied in addition to PageRank. *Technology Overview*, GOOGLE, <http://www.google.com/about/company/tech.html> (last visited Feb. 25, 2012).

⁸⁷ *Google History*, GOOGLE, <http://www.google.com/about/company/history.html> (last visited Feb. 25, 2012).

⁸⁸ U.S. Patent No. 6,285,999 col. 1 ll. 51–67 (filed Jan. 9, 1998).

⁸⁹ *See id.*

⁹⁰ *Id.*

⁹¹ *Id.* col. 2 ll. 51–67, col. 3 ll. 1–3.

⁹² *Id.* col. 3 ll. 4–10.

⁹³ *Id.* col. 8 ll. 55–67.

⁹⁴ *See supra* section I.B.

specified, efficient algorithm.⁹⁵ Under current law, these definitions of the invention in the specification limit claim scope.⁹⁶

Page's discovery was *how* to rank pages using incoming links, not that pages could be ranked by using incoming links at all. Therefore, to claim this method of how to do something, Page had to make a detailed disclosure of the formula and an algorithm for how to calculate it.⁹⁷ Consequentially, the PageRank patent has similarities to patents on physical apparatuses in that it can be designed around. Future inventors can come up with a new formula for ranking pages using incoming links and not infringe the patent.

Nonetheless, the PageRank patent also illustrates how a scientifically advanced software patent is different from a physical object patent. First, the potential design-around of the patent is limited to the aspects operating on the edge of mathematical knowledge. For example, the design-around of PageRank is limited to a single aspect, the formula. For a physical apparatus there would likely always be many possible design-arounds, such as using different materials, different components, different structural layouts, and so on, because all of these characteristics operate on the edge of physical knowledge. Second, a scientifically advanced software patent could cover the single best way to accomplish a function, thereby meaning that all remaining design-arounds are worse. In mathematics, it is possible to prove that an algorithm is the single most efficient method of accomplishing a function.⁹⁸ On the other hand, a given physical object cannot be categorically proven to be the best possible for performing a function. The PageRank formula could be suspected of being the best possible way of ranking pages according to incoming links because it is based on a sound mathematical intuition — pages can be ranked based on the steady-state probability distribution of a web surfer who follows web links randomly.⁹⁹ The fact that the PageRank formula has a strong mathematical justification suggests that formulas that vary from it would likely be illogical.

Therefore, scientifically advanced software patents that implement complex algorithms or new mathematical principles exist in a middle ground between “business method”-type software patents and physical object patents. They may offer a small opening for design-arounds by reciting a precise way of performing a function that is improvable.

⁹⁵ '999 Patent col. 4–6.

⁹⁶ See *Honeywell Int'l, Inc. v. ITT Indus., Inc.*, 452 F.3d 1312, 1318 (Fed. Cir. 2006).

⁹⁷ 35 U.S.C. § 112, para. 1 (2006).

⁹⁸ See, e.g., STEVEN S. SKIENA, *THE ALGORITHM DESIGN MANUAL* 130 (2d ed. 2008) (proving that currently known sorting algorithms are the fastest possible).

⁹⁹ '999 Patent col. 5 ll. 21–23.

II. PAST ATTEMPTS TO DISTINGUISH SOFTWARE PATENTS

Commentators have largely agreed that software patent claims tend to be broader than physical object patent claims. One prominent explanation is that the distinction is due to an accident: software lends itself to abstract claim language, though any product could be patented in similarly abstract terms. Another explanation posits that courts have allowed the distinction to persist because they overestimate the skill of the ordinary software programmer. Both theories, however, fail to offer a compelling explanation for the breadth of software patents because of their refusal to acknowledge the key characteristic of software: that it is inherently functional.

James Bessen and Professor Michael J. Meurer are proponents of the first explanation: software patents are different from physical object patents because software is an “abstract” technology that lends itself to abstract claim language.¹⁰⁰ By “abstract,” they mean that it is difficult to map words in a patent to actual technologies.¹⁰¹ Nonetheless, they believe that “[a]ny technology can be claimed abstractly.”¹⁰² In their view, abstract language causes both patent breadth and uncertainty in coverage, and these factors in combination lead to ambush litigation.¹⁰³

Bessen and Meurer argue that abstract claim language leads to patent breadth because it allows inventors to patent inventions they did not invent.¹⁰⁴ In other words, inventors “overclaim.”¹⁰⁵ An inventor who develops a single instantiation of an invention might use abstract language to cover many similar and related inventions that achieve the same result, though she did not actually invent them.¹⁰⁶ Bessen and Meurer’s conclusion that inventors are overclaiming would imply that the courts and the U.S. Patent and Trademark Office are applying an enablement and written description requirement that is too lenient — otherwise all these “abstract” patents would be invalid.¹⁰⁷

In addition, according to Bessen and Meurer, abstract claim language leads to uncertainty because (1) it is difficult to determine what an abstract patent covers just by reading the claims, and (2) it is unclear whether the patent will be given a narrow or broad interpreta-

¹⁰⁰ BESSEN & MEURER, *supra* note 2, at 22–23, 187.

¹⁰¹ *Id.* at 22.

¹⁰² *Id.* at 213; *see also id.* at 66–67 (describing what the authors believe to be overly broad claiming of a biotechnology invention).

¹⁰³ *Id.* at 47.

¹⁰⁴ *Id.* at 199–200.

¹⁰⁵ *Id.* at 66–67, 213.

¹⁰⁶ *Id.* at 66–67, 213 (“[T]oo many software patents claim all technologies with similar form or all means of achieving a result, when the actual invention is much more limited and often trivial.” *Id.* at 213.)

¹⁰⁷ *Id.* at 210.

tion in litigation.¹⁰⁸ As a result of the uncertainty, a company could easily infringe the patent without realizing it.¹⁰⁹ It may know of the patent but have a different view of what it covers.¹¹⁰

On the other hand, Professors Dan L. Burk and Mark A. Lemley argue that software patents are broader than physical object patents because the Federal Circuit is applying a PHOSITA¹¹¹ standard that mistakenly assumes individuals working with software are more highly skilled than individuals working with other technologies.¹¹² In their view, the Federal Circuit's rule that disclosure of the functionality of a program, without source code or other guidance, is enough to support a software claim rests on an unrealistic assumption that programmers are incredibly skilled.¹¹³ Because of this assumption, software patents are broader than nonsoftware patents — the inventor can claim more because she is less limited by the requirements for enablement and written description.¹¹⁴ Once again, this is a statement about overclaiming by patentees. In other words, Burk and Lemley's point is that an overly relaxed PHOSITA standard creates broad software patents because it enables overclaiming.¹¹⁵

Both sets of authors recognize that software patents can be claimed at a high conceptual level. Bessen and Meurer see it as a consequence of the use of abstract language that is equally applicable to physical object patents as well.¹¹⁶ Burk and Lemley see it as a result of the lack of a legal requirement for implementation details of software.¹¹⁷

In contrast with the above authors, this Note asserts that software patents are broad without being overclaimed. Software patents are broad not because designers are clever writers of abstract claims or because they are overclaiming their expertise; they are broad because the patentable discovery in software is function unmoored from any secondary characteristics.

Whether software is in fact easier to understand than physical objects, or whether software designers overclaim their expertise, is, in part, an empirical question. But it is unlikely to be answered. To judge whether an inventor has overclaimed requires knowing the “right” scope of coverage she should be allowed based on the enable-

¹⁰⁸ *Id.* at 199–200.

¹⁰⁹ *Id.* at 47, 67.

¹¹⁰ *Id.*

¹¹¹ Burk & Lemley, *supra* note 42, at 1185–86.

¹¹² *Id.* at 1191–92; *see also id.* at 1163–67, 1170–71.

¹¹³ *Id.* at 1191–92.

¹¹⁴ *Id.* at 1170–71.

¹¹⁵ *Id.*

¹¹⁶ BESSEN & MEURER, *supra* note 2, at 22, 213.

¹¹⁷ Burk & Lemley, *supra* note 42, at 1170–71.

ment and written description.¹¹⁸ That is a highly uncertain question, not just in terms of a legal determination under current doctrine,¹¹⁹ but also in terms of the factual question of what the inventor *actually invented* and therefore deserves rights to.¹²⁰ Thus, how often overclaiming occurs is not subject to easy empirical proof.¹²¹

Furthermore, the view that inventors are claiming too much is potentially inconsistent with the fact that the majority of patents asserted by NPEs originally came from productive companies.¹²² These companies would seem to have low incentives to aggressively overclaim. Research shows that NPEs win only 9.2% of their cases and software patentees only 12.9%,¹²³ the high loss rate coming from a mix of invalidations and findings of noninfringement.¹²⁴ While the invalidations could be evidence of overclaiming, it would seem odd for productive companies to seek and obtain software patents that are so overly broad as to be invalid, rather than covering their own product.¹²⁵ What could instead be happening is that NPEs are obtaining from productive companies patents that are actually broad — due to the functional nature of software — but the patents are invalid, perhaps under inherent anticipation¹²⁶ or due to little-known prior art,¹²⁷ because the inventors did not contemplate what wide scope their inventions actually had.¹²⁸

Nonetheless, this Note's thesis is not inconsistent with an account that some overclaiming occurs. Software patents could be broad both because of software's inherent qualities — as this Note argues — and because of inventors' claiming too much on top of such natural

¹¹⁸ Allison et al., *supra* note 1, at 707.

¹¹⁹ See *Arlington Indus., Inc. v. Bridgeport Fittings, Inc.*, 632 F.3d 1246, 1257 (Fed. Cir. 2011) (Lourie, J., concurring in part and dissenting in part); Oskar Liivak, *Rescuing the Invention from the Cult of the Claim* 3, 5 (Feb. 24, 2011) (unpublished manuscript), available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1769270.

¹²⁰ See *Arlington Indus.*, 632 F.3d at 1257–58 (Lourie, J., concurring in part and dissenting in part); Liivak, *supra* note 119, at 6–8.

¹²¹ Allison et al., *supra* note 1, at 707.

¹²² Risch, *supra* note 10, at 27–28.

¹²³ Allison et al., *supra* note 1, at 680–81.

¹²⁴ *Id.* at 706–07.

¹²⁵ Studies have shown that, while productive companies value patents both for protecting their own products and for their blocking function, they value the protective function more. Cohen et al., *supra* note 24, at 17–18.

¹²⁶ Inherent anticipation is when prior art contains the invention, but the invention was not recognized by the prior inventors. See *In re Schreiber*, 128 F.3d 1473, 1479–80 (Fed. Cir. 1997).

¹²⁷ Invalidating prior art could easily exist without the knowledge of the inventors. Patents and printed publications anywhere in the world are all prior art, as are any public uses in the United States. 35 U.S.C. § 102 (2006).

¹²⁸ If, however, overclaiming were the reason for breadth, prior art invalidations would be less likely. The patentee could just claim an invention far into the future that no one had enabled yet. Overclaiming would allow the patentee to claim the invention before prior art exists.

breadth. The point is that software patents are broad even when they legitimately claim only the actual invention.

III. IMPLICATIONS

Having set forth this Note's argument about how and why software patents are different from physical object patents, this Part discusses the implications of those conclusions.

A. Ambush Litigation

It is the functional nature of software and lack of design-arounds that make software patents more effective for NPEs to raise in ambush litigation. These factors make software patents less likely to obsolesce. As time goes by, the conceptual language used in the software patent captures more after-arising technology that develops.¹²⁹ When the patented invention is functionality itself, it will cover later ways of implementing that functionality because the implementation details are not part of the invention.¹³⁰

Freeny's patent, for example, was for a method of doing things, not for a specific implementation of that method. Freeny described every object in his claims using conceptual terminology. For instance, he posited an "information manufacturing machine" (IMM) that would produce a material object in response to commands from a user.¹³¹ Given that Freeny did not limit the IMM to a specific physical apparatus, the IMM could be any device that performed the requisite functions — including a personal computer. The patent failed to cover e-commerce only because Freeny had described the IMM as producing a material object that the customer could take home.¹³²

Some commentators have suggested that it would be unfair for Freeny's patent to cover e-commerce because Freeny admitted that he did not foresee the World Wide Web,¹³³ but their reasoning is fallacious. Freeny's claim involved communication steps such as "providing . . . the information to be reproduced," "providing a request reproduction code," and "providing an authorization code."¹³⁴ It is irrelevant whether the communication would occur over a direct electronic modem connection, as disclosed in Freeny's specification, or instead be routed through a series of switches comprising the internet

¹²⁹ BESSEN & MEURER, *supra* note 2, at 67.

¹³⁰ This may be an alternative way of viewing what Bessen and Meurer deem to be claim terms in software patents "chang[ing] in meaning over time." *Id.* (emphasis omitted).

¹³¹ See *supra* notes 4–8 and accompanying text.

¹³² *Interactive Gift Express, Inc. v. CompuServe Inc.*, 256 F.3d 1323, 1338 (Fed. Cir. 2001).

¹³³ Fahmi & Dreszer, *supra* note 4, at 318.

¹³⁴ U.S. Patent No. 4,528,643 col. 28 ll. 22–47 (filed Jan. 10, 1983).

before reaching its end points.¹³⁵ The implementation details of how communication would occur have nothing to do with Freeny's patentable invention. Freeny's invention was solely a discovery of a method and had nothing to do with specific physical technology. Doctrinally, Freeny's patent *should* cover products that use the World Wide Web.

Furthermore, there is no evidence that Freeny was attempting to overclaim. His goal was not to obtain a broad patent to block others, but instead to cover a real product that he was developing.¹³⁶ However, his actual invention, once the implementation details were taken out, was similar to e-commerce.

B. *How Should Software Patents Be Treated?*

This Note's thesis that software is inherently different from other kinds of technology may have implications for how software should be treated in the patent system.

First, it suggests that judicial attempts to limit broad software claims by reading in limitations from the specification are misguided.¹³⁷ One Federal Circuit judge has gone so far as to call the specification "the heart of the patent."¹³⁸ But this view fundamentally misunderstands what a patent covers. In a software patent, the *actual invention* is functionality. If judges try to limit software patents to implementation details described in the specification, their choices of implementation will be arbitrary.

Second, one may ask whether, if software patents tend to cover many or all ways of accomplishing functionality, software patents should be limited in some way or whether software should be patentable at all. These are complex policy questions that this Note does not aspire to answer. The Note only seeks to show that software is "different" in a way that affects how software plays out in the patent system.

Nonetheless, this Note's observations should not be taken as an excuse to cut apart the software patent regime. Given the prior voracity of the attacks on software patents in the absence of clear reasoning, it is unclear whether software patents can still get a fair hearing. There is no clear empirical evidence of how much of a problem ambush liti-

¹³⁵ See *SiRF Tech., Inc. v. Int'l Trade Comm'n*, 601 F.3d 1319, 1329–30 (Fed. Cir. 2010) (construing the term "communicat[ing]" to include communication over a series of links (alteration in original)).

¹³⁶ See Fahmi & Dreszer, *supra* note 4, at 317.

¹³⁷ For an example of a court reading in such limitations, see the lower court ruling for the defendants in *Interactive Gift Express, Inc. v. CompuServe Inc.*, 47 U.S.P.Q.2d (BNA) 1797, 1809–10 (S.D.N.Y. 1998), *vacated*, 231 F.3d 859 (Fed. Cir. 2000), *opinion withdrawn and superseded on reh'g in part*, 256 F.3d 1323 (Fed. Cir. 2001).

¹³⁸ *Arlington Indus., Inc. v. Bridgeport Fittings, Inc.*, 632 F.3d 1246, 1257 (Fed. Cir. 2011) (Lourie, J., concurring in part and dissenting in part).

gation of software patents is.¹³⁹ These suits cost large companies money, but there is no clear evidence that they disincentivize investment, research, or advances in technology in a way that damages society as a whole.¹⁴⁰

The patent system provides an incentive for inventors to create new technologies that would not otherwise be developed, or whose development would otherwise be delayed. There may be a cognitive bias against software patents because individual cases of blocking are easier to detect than the positive, systemic effects on innovation as a whole. Cutting back on the software patent regime risks cutting back on many innovative, good patents in addition to the potentially bad ones.¹⁴¹ Although software is different from other types of inventions, it has always been the role of the patent system to adapt to new and important technologies.¹⁴²

CONCLUSION

Software patents differ from physical object patents in that they do not recite physical characteristics or other implementation details. The reason for the difference is the dichotomy between the well-understood mathematical world of software, which is easy to generalize, and the messy physical world, which is hard to generalize. This inherent difference is important in guiding understanding of what the actual invention is in a software patent versus a physical object patent. The software invention is just a function. Once this aspect of software inventions is understood, the common appearance of software patents in litigation is not surprising.

¹³⁹ Risch, *supra* note 10, at 2.

¹⁴⁰ *Id.* But see BESSEN & MEURER, *supra* note 2, at 91–94, 144–46.

¹⁴¹ Edward Van Gieson & Paul Stellman, *Killing Good Patents to Wipe Out Bad Patents: Bilski, the Evolution of Patentable Subject Matter Rules, and the Inability to Save Valuable Patents Using the Reissue Statute*, 27 SANTA CLARA COMPUTER & HIGH TECH. L.J. 403, 403–04 (2011).

¹⁴² See *Bilski v. Kappos*, 130 S. Ct. 3218, 3227–28 (2010).