

Marking the Software Patent Beast

Stephen Lindholm[†]

The literature of software patents has thus far tried to directly address whether software patents increase innovation. The wholesale reform papers have persuaded neither the courts nor Congress, perhaps due to the unfortunate dearth of economic data.

This paper starts from the proposition that software patents are, practically speaking, hidden away in the recesses of the patent office and practically impossible to find. It proceeds under the first economic principles of the patent system to argue that there can be no justification for patenting software when the public has no knowledge of the patents' scope or technical disclosure. It concludes by observing that patent law already provides a mechanism for disclosing patents to the public, the marking duty, and proposes putting teeth into it so that holders of software patents would be required to play by the same rules as holders of other kinds of patents.

The likely effect of a strengthened duty to mark would be to neutralize the great numbers of software patents that their owners would not deem worth the legal fees to check against their products and the products of their licensees. It would discourage patent holders from expansively re-interpreting their patents years after the fact, as well as ameliorating the blanket license "tax" large patent holders levy on smaller companies. Yet any economic benefits of software patents would be preserved or even enhanced under a stronger marking duty.

One hundred thousand software patents are in force today, yet nobody really knows what is covered or by whom. Patents on computer software are so obscure as to be effectively secret—they are abstrusely written, not indexed in any meaningful way, and their scope is hard to predict. Because the economic principles supporting the American patent system depend on patents being publicly known, this obscurity undermines the economic justification for software patents.

A solution to the software patent obscurity problem may lie in the disused "marking duty." In theory, patent owners have a duty to label their products with the relevant patent numbers. The duty to mark is justified by patent owners being

[†] Stephen Lindholm, Stanford University M.S. 2002, J.D. 2005. The author thanks Tony Reese for his insight and guidance. This article is hereby released into the public domain, effective five years from publication in this Journal.

best situated to determine the scope of their patents. Competitors and the public can determine which patents cover what by inspecting marked goods.

This article is divided into three parts. The first part briefly introduces the principles behind the U.S. patent system and discusses the legal development of software patents in this country. The second part discusses the obscurity problems encountered with software patents. The third part introduces the marking duty of patent law and explains how it could be better used to mitigate the software patent obscurity problem.

I. An introduction to software patents

This Part describes the principles behind the U.S. patent system, as well as the development of the law regarding software patents in this country, as a way of framing the discussion later in this article. We will see in subsequent parts of the article that there are problems administering the body of extant software patents, and it is useful to show that the proposed solution to these problems is designed to further existing patent law, rather than being a veiled attempt to change the law, benefiting one group at the expense of another.

A. Patents as exceptions to a rule against monopolies

The American patent system dates from the first years of the country, and then, as now, economic thinkers justified patent monopolies as incentives for inventors. In particular, they did not necessarily think that inventors had a natural right to their ideas. To defend patents as a matter of public policy, they had to be shown to actually encourage innovation.

The architects of this nation's government, in allowing for patents on inventions, made an exception to their general dislike of government-protected monopolies. As they were intimately familiar with the Crown's abuses of monopolies,¹ the founders were cautious of granting Congress the same power.² In fact, several ratifying conventions proposed prohibiting government monopolies

¹ Royal monopolies included the exclusive rights to make basic essentials or control existing businesses. These monopolies were, for the most part, sold for cash or handed out to courtiers without regard for the benefit of society. BRUCE W. BUGBEE, *GENESIS OF AMERICAN PATENT AND COPYRIGHT LAW* 37 (1967). The colonists were of course well-acquainted with the abuses committed by the monopolies themselves, such as the British East India Tea Company. *Id.*

² A distrust of monopolies spanned colonial politics. "With regard to monopolies they are justly classified among the greatest nuisances in Government." Letter from James Madison to Thomas Jefferson (Oct. 17, 1788). *See also* Letter from Thomas Jefferson to James Madison (July 31, 1788), in 7 *THE WRITINGS OF THOMAS JEFFERSON* 98 (1907).

altogether.³ The Constitution emerged as allowing a subset of monopolies in the form of copyrights and patents.⁴

It is a little unclear what motivated adding the statutory monopoly clause. Was it intended to protect the “natural rights” of authors and inventors? The word “secure” at least suggests that copyright and patents only affirm preexisting natural rights of possession to authors and inventors.

However, in 18th century England, patents were absolutely not seen as a natural right,⁵ and it is at best debatable whether copyright was.⁶ It seems more likely that the founders, familiar with the societal hardships of too many monopolies,⁷ wanted to limit monopolies to where they would benefit society. This “utilitarian” outlook was espoused in the clause’s preamble:⁸ “To promote the Progress of Science and useful Arts.” Under this view patents are allowed only insofar as they encourage technological progress. Thinkers like Thomas Jefferson squarely rejected the notion of a natural right in ideas, contrasting property against

³ A constitutional ban on monopolies was considered several times. The Massachusetts ratifying convention proposed: “That Congress erect no company with exclusive advantages of commerce.” 2 JONATHAN ELLIOT, *THE DEBATES IN THE SEVERAL STATE CONVENTIONS ON THE ADOPTION OF THE FEDERAL CONSTITUTION, AS RECOMMENDED BY THE GENERAL CONVENTION AT PHILADELPHIA, IN 1787*, at 177 (2d ed. 1876). The ratifying conventions of New Hampshire, New York, North Carolina, and Rhode Island also proposed amendments. HERMAN V. AMES, *THE PROPOSED AMENDMENTS TO THE CONSTITUTION OF THE UNITED STATES DURING THE FIRST CENTURY OF ITS HISTORY* 255 (Washington, Gov’t Printing Office 1897). Three attempts to introduce the amendment through Congress also failed. *Id.*

⁴ “[Congress shall have Power] To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries.” U.S. CONST. art. I, § 8, cl. 8.

⁵ English patents were granted by grace and favor, not as a matter of right. Edward C. Walterscheid, *To Promote the Progress of Science and Useful Arts: The Background and Origin of the Intellectual Property Clause of the United States Constitution*, 2 J. INTELL. PROP. L. 1, 13 n.40 (1994) (citing W.M. HINDMARCH, *A TREATISE ON THE LAW RELATIVE TO PATENT PRIVILEGES FOR THE SOLE USE OF INVENTIONS* 3 (Harrisburg, Pa., I.G. M’Kinley & J.M.G. Lescure 1847) (1846)).

⁶ Copyright may have been viewed as a natural right by some, see LYMAN RAY PATTERSON, *COPYRIGHT IN HISTORICAL PERSPECTIVE* 194–95 (1968), but even that was not universal.

⁷ See BUGBEE, *supra* note 1.

⁸ The language in clause eight may have been a compromise, or perhaps, as the clause was inserted in the waning days of the Constitutional Convention, the language was not as carefully considered as the rest of the Constitution. According to James Madison’s notes, the proposal to include patent and copyright powers was unanimously approved on August 18 with no debate; the wording of the clause was left to committee. That very day, delegate James Rutledge “remarked on the length of the Session, the probable impatience of the public and the extreme anxiety of many members of the Convention to bring the business to an end.” Less than a month later the Convention finished. JAMES MADISON, *DEBATES IN THE FEDERAL CONVENTION OF 1787*, at 480 (Ohio University Press 1984).

what some today label “intellectual property.”⁹ This philosophy accords with modern rulings by the Supreme Court.¹⁰

But *could* patents encourage innovation? This was an open empirical question for the framers. Early skeptics of the patent system like Thomas Jefferson at first doubted that “ingenuity [could be] spurred on by the hope of a monopoly for a limited time.”¹¹ But after a few years even Jefferson changed his mind. He believed that some of the inventions brought to the public were valuable enough to justify the costs of the entire patent system.¹² He presumably thought that, in the absence of the patent system’s incentives, these inventions would have been kept secret or would not have been brought to practice in the first place. The patent system, according to Jefferson, could be rationalized as utilitarian.

One might ask if the utilitarian framework is still the proper way to analyze the patent system more than two centuries after men in powdered wigs wrote the Constitution. Patent litigation has become a high-roller’s game, with huge legal fees and verdicts.¹³ Is it possible to measure the benefits and costs of the patent system accurately enough to justify major changes? Alternatively, if, as it appears, the Lockean idea that labor deserves a property right has taken root instead, is it not just to secure the rights of authors and inventors despite any costs to society?

Natural rights have come back into vogue over the past few decades through the neologism “intellectual property.” This rhetorical shift has already begun in the context of copyright. Copyright holders now frame their arguments as what they should be entitled to as property owners, rather than what is best for the creative

⁹ Letter from Thomas Jefferson to Issac McPherson (Aug. 13, 1813), in 13 THE WRITINGS OF THOMAS JEFFERSON 333 (1903). Thomas Jefferson was not present at the Constitutional Convention.

¹⁰ The modern Supreme Court has indicated that Jefferson’s utilitarian philosophy on the patent system, *see infra* note 12, is entitled to deference. *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 7 (1966). He served as the *de facto* first head of the U.S. patent office under George Washington.

¹¹ Letter from Thomas Jefferson to James Madison (July 31, 1788), in 7 THE WRITINGS OF THOMAS JEFFERSON 98 (1907).

¹² “An act of Congress authorizing the issuing of patents for new discoveries has given a spring to invention beyond my conception. Being an instrument in granting the patents, I am acquainted with their discoveries. Many of them indeed are trifling, but there are some of great consequence, which have been proved by practice, and others which, if they stand the same proof, will produce great effect.” Letter from Thomas Jefferson to Benjamin Vaughan (June 27, 1790), in 8 THE WRITINGS OF THOMAS JEFFERSON 50 (1907).

¹³ For patent litigation fees, *see infra* note 132. To use Microsoft as an example of patent litigation damages, in the past two years it has paid out mind-numbing billions in patent suits: to Eolas, \$565m; to InterTrust, \$440m; to ImagExpo, \$62m; to Immersion, \$35m; to Sun Microsystems, \$1,250m plus future royalties. There have also been undisclosed settlements with AT&T, Kodak, and E-Data. Doubtless other suits settled for confidential amounts, as Microsoft faces about 30 patent suits at any given time. These figures do not include antitrust damages or other commercial litigation. Microsoft Corp., *S.E.C. 10-K Filing for Fiscal Year Ended June 30, 2004*, II-8 n.17 (filed Sept. 9, 2004).

and artistic progress of society.¹⁴ More generally, there has been a shift away from open and free competition as the baseline principle of the American economic and legal systems.¹⁵

The response to both objections is to point out the sheer cost of the software-patenting regime and the higher footing onto which software patents have glommed. If offsetting benefits cannot be shown by the system's proponents, at the very least software patent owners should be required to obey the same disclosure rules as other patent holders.

B. Economic analysis of patent disclosure

Patents can motivate invention, disclosure, commercialization, and designing around,¹⁶ as well as provide a solution to Kenneth Arrow's information paradox. If, as Thomas Jefferson believed, the patent files are mostly deadwood with a few valuable patents here and there,¹⁷ the natural way to begin trimming the deadwood is to analyze the ways that patents can contribute to technological progress. These incentives are explained in more detail in this section.

These incentives are less effectual with computer software patents, owing to network externalities, licensing practices, and standardization.¹⁸ They are not at all effective when the contents of patents are kept secret. Both the technical information described in the patents and the scope of the patent monopolies must be known by competitors.

In Part Two it is argued that the contents of software patents are not known by competitors or even by the patent owners. In Part Three it is shown how a

¹⁴ Judge Noonan of the Ninth Circuit, at the oral argument of *Metro-Goldwyn-Mayer Studios, Inc. v. Grokster, Ltd.*, had to admonish the plaintiff to argue its case on the merits:

Let me say what I think your problem is. You can use these harsh terms, but you are dealing with something new, and the question is, does the statutory monopoly that Congress has given you reach out to that something new. And that's a very debatable question. You don't solve it by calling it "theft." You have to show why this court should extend a statutory monopoly to cover the new thing. That's your problem. Address that if you would. And curtail the use of abusive language.

Oral Argument, *Metro-Goldwyn-Mayer Studios, Inc. v. Grokster, Ltd.*, 380 F.3d 1154 (9th Cir. 2004) (Nos. 03-55894, 03-55901, 03-56236), available at http://www.eff.org/IP/P2P/MGM_v_Grokster/20040203_oral_arg.mp3.

¹⁵ Robert P. Merges, *One Hundred Years of Solicitude: Intellectual Property Law, 1900-2000*, 88 CAL. L. REV. 2187, 2239-40 (2000).

¹⁶ See generally Rebecca S. Eisenberg, *Patents and the Progress of Science: Exclusive Rights and Experimental Use*, 56 U. CHI. L. REV. 1017 (1989).

¹⁷ As author Theodore Sturgeon later became famous for saying, "Ninety percent of everything is crud." Theodore Sturgeon, *Books: On Hand*, VENTURE SCIENCE FICTION, Mar. 1958, at 66-67 (book review).

¹⁸ This distinction, if empirically strong enough, might undercut the rationale for a software-patenting regime altogether.

strengthened marking requirement would address the obscurity problem and put software patents on the same footing as other kinds of technology.

1. Incentive to invent

Inventors in the absence of a patent system may capture only a fraction of their idea's benefits. Developing an idea may be too costly to make research attractive, even if on the whole it would provide a net economic benefit to society.¹⁹ In the absence of a patent system, competitors may copy ideas from an originator, whose profits then decline along with his prices and market share. The pioneer, in effect, is stuck with the costs of blazing the trail.

Patents encourage invention by giving inventors money for their ideas. They can sell or license their patents outright for cash, or start a business and use the patent to exclude competition. Patent income is tightly correlated with the usefulness of the invention claimed in the patent.²⁰ Bright ideas can be extracted from brilliant minds by awarding patents, assuming that one accepts Stanford computer science professor Jeff Ullman's proposition that money is the best motivator for researchers.²¹

It is important to publicize patented inventions so that later inventors do not waste time re-discovering old inventions. Inventors who think they won't be rewarded for their success will not be motivated to do research.²² Congress long ago determined that the best way to disseminate patented research is to require a detailed, enabling disclosure in the patent itself. Anybody can look up the

¹⁹ Eisenberg, *supra* note 16, at 1024–28.

²⁰ The patent may be broader than what the inventor actually reduced to practice. Whether this is economically beneficial (the “prospect theory”) has been thoroughly discussed in the literature and may depend on the field of technology. Compare Edmund W. Kitch, *The Nature and Function of the Patent System*, 20 J.L. & ECON. 265 (1977) (asserting that broad patent grants are beneficial to society, by analogy to mineral rights), with Mark A. Lemley, *The Economics of Improvement in Intellectual Property*, 75 TEX. L. REV. 989, 1048–57 (1997) (showing Kitch's prospect theory to be bunkum); Lawrence Lessig, *Exclusive Rights to Stagnate*, FIN. TIMES, Feb. 20, 2003 (arguing software patents choke innovation by covering too much); James Bessen & Eric Maskin, *Sequential Innovation, Patents, and Imitation* (2000) (Department of Economics, Massachusetts Institute of Technology, Working Paper No. 00-01, 2000) (economic and empirical analysis shows that industries with complementary innovation, such as computer software, are impeded by patents).

²¹ “Communism—an intuitively appealing concept—failed because it forgot that the best minds need to be motivated, and money works better than anything else, on average.” Prof. Jeffrey D. Ullman, 2000 Knuth-Prize Lecture (Nov. 16, 2000), available at <http://www-db.stanford.edu/~ullman/pub/focs00.html> (last visited Apr. 15, 2005). The alternatives would presumably include prizes, prestige, jobs, etc. His speech was against the permissive over-patenting of software.

²² The principle that a later re-inventor deserves a patent over someone who hides his ideas is found elsewhere in patent law, in something called an “interference proceeding.” Interference proceedings determine which of two or more inventors should receive a patent. Generally, the one who earlier reduced the invention to practice will get the patent, unless he abandoned, suppressed, or concealed it. 35 U.S.C. § 102(g) (2005).

information in patents for free, either at one of dozens of U.S. patent libraries or on the Patent and Trademark Office web site. The natural limitation is, of course, that someone interested in a certain kind of research must be able to find the patent among the over six million on file. Software patents are, in fact, notably difficult to find.²³

Likewise contrary to the theory of encouraging research are “harvested” patents, or those that are applied for as an incidental by-product of ordinary research and development.²⁴ This ordinary kind of research is not motivated by the prospect of a patent. If an invention is so easily rediscovered that no incentive is needed to develop it in the first place, then society suffers a deadweight loss without economic gain.²⁵

It was never the object of those laws to grant a monopoly for every trifling device, every shadow of a shade of an idea, which would naturally and spontaneously occur to any skilled mechanic or operator in the ordinary progress of manufactures. Such an indiscriminate creation of exclusive privileges tends rather to obstruct than to stimulate invention. It creates a class of speculative schemers who make it their business to watch the advancing wave of improvement, and gather its foam in the form of patented monopolies, which enable them to lay a heavy tax upon the industry of the country, without contributing anything to the real advancement of the art. It embarrasses the honest pursuit of business....²⁶

Perhaps one believes that harvested patents are “obvious,” but the standard for obviousness in patent law does not go so far.²⁷ The prime reason that large companies harvest patents is that it increases blanket-licensing revenue. If the law were changed to impose practical difficulties on blanket-licensing software patent portfolios, it would indirectly curtail patent harvesting.

2. Incentive to design around

Patents indirectly encourage inventors to find multiple solutions to problems. A patent grants the original inventor the exclusive right to practice his

²³ The practical obscurity of software patents is argued in Part Two.

²⁴ See, e.g., Steven I. Weisburd, *Handling Intellectual Property Issues in Business Transactions*, 690 PLI/PAT 39 (2001) (discussing benefits of “extracting value” from harvested patents, none of which include justifying research which would otherwise be unprofitable).

²⁵ “Preventing independent discovery is a side effect of the patent system, not its goal.” Dan L. Burk & Mark A. Lemley, *Policy Levers in Patent Law*, 89 VA. L. REV. 1575, 1585 n.24 (2003).

²⁶ *Great Atl. & Pac. Tea Co. v. Supermarket Equip. Corp.*, 340 U.S. 147, 155 (1950) (quoting *Atl. Works v. Brady*, 107 U.S. 192, 200 (1883)) (Douglas, J., concurring).

²⁷ The reason is that, as Jefferson observed, the patent system is designed to benefit society by granting many trivial patents alongside a handful of good ones. Patent harvesting disproportionately increases the number of trivial patents.

idea. If he chooses to exercise that right, competitors need to find alternative, unpatented solutions to enter or remain in the market. This incentive for competitors to conduct further research is called the incentive to “design around,” and is one of the ways the patent system leads to advances in technology.²⁸ Examples of such advances include the ulcer drug Zantac, made by designing around patents on the older drug Tagamet, and Xerox’s photocopying technology, made by designing around Kodak’s silver halide patents.²⁹

The incentive to design around is a way for competitors to build on the research of the first inventor. They seek out ways to achieve a specific goal not claimed by the original researcher’s patent. It is axiomatic that if a researcher sets out to design around a patent, he must first know what the scope of the patent claims are, and further, he may benefit greatly from the technical information disclosed in the patent.³⁰ It is, therefore, necessary that follow-on researchers be able to find patents in their area of research.

Whether follow-on innovation is generally worth the costs is an open empirical question. There are people who argue against and for benefits of designing around patents. Carl Shapiro notes that the costs to society of designing around a patent are particularly high when, as is often the case, patents are improperly granted.³¹ He also observes that designing around inevitably leads to legal costs, which do not benefit society.³² Others assert that designing around has a net benefit.³³

The incentive to design around is inherently limited because inventors have every incentive to draft their patent claims broadly,³⁴ and in those areas competitors

²⁸ *Slimfold Mfg. Co. v. Kinkead Indus., Inc.*, 932 F.2d 1453, 1457 (Fed. Cir. 1991).

²⁹ FED. TRADE COMM’N, TO PROMOTE INNOVATION: THE PROPER BALANCE OF COMPETITION AND PATENT LAW AND POLICY, ch. 2, at 21 (2003), available at <http://www.ftc.gov/os/2003/10/innovationrpt.pdf>.

³⁰ Stephen G. Kunin & Linda S. Therkorn, *Patent Issues Likely to Directly Affect The Development Of The Agricultural and Microbial Biotech Industry Over The Next Five Years*, 86 J. PAT. & TRADEMARK OFF. SOC’Y 501, 502 (2004). The other way for competitors to build on the research is to license the original invention and do research in the same area. *Id.*

³¹ Nearly half of all litigated patents are invalidated. John R. Allison & Mark A. Lemley, *Empirical Evidence on the Validity of Litigated Patents*, 26 AIPLA Q.J. 185, 205 (1998). There is probably a bias in that weaker patents are more likely to be contested.

³² Carl Shapiro, *Patent System Reform: Economic Analysis and Critique*, 19 BERKELEY TECH. L.J. 1017, 1019 (2004).

³³ *FTC/DOJ Hearings on Competition and Intellectual Property Law and Policy in the Knowledge-Based Economy* (statement of Frederick J. Telecky, Jr., Senior Vice President and General Patent Counsel, Texas Instruments), available at <http://www.ftc.gov/opp/intellect/020228telecky.pdf>. Texas Instruments is famous for its aggressive patent licensing strategy, and so perhaps it is not surprising that it asserts that patents do not impede research by competitors in the area. See *infra* note 134.

³⁴ Paul M. Janicke, *When Patents Are Broadened Midstream: A Compromise Solution to Protect Competitors and Existing Users*, 66 U. CIN. L. REV. 7, 13 (1997).

are prohibited from researching many alternative solutions.³⁵ Competitors are chilled from research when they believe the outcome might infringe a patent.

It is often impossible to tell in advance whether a subsequent researcher's use of a patented invention will lead to an improvement falling within the scope of the claims of the prior patent or to a substitute technology falling outside the patent claims. The uncertainty arises in part because it is difficult to predict the course and outcome of research projects, and in part because it is difficult to determine the validity and scope of patent claims until these matters are resolved in litigation.³⁶

Competitors may be unable to research even unpatented technology if infringement would be required during the course of research. There once was a common-law exception for experimental use of an invention, but little of it remains.³⁷ In any case, there is no empirical evidence on how much of an economic benefit comes from the incentive to design around, other than the handful of anecdotes related above.

There is, in fact, an argument that designing around patents is *counterproductive* with computer software and other network-effect technologies,³⁸ unless competitors are able to design around the patents quite early.³⁹ It may be technologically impossible to develop an alternative that both is compatible with the existing network and falls outside the scope of the patent.⁴⁰ A new, incompatible design that splits the network may be inherently less desirable. The older, patented technology will become less useful for its users, and the new technology will not achieve the same usefulness even if it is an improvement over the older technology. It is probably the case that competitors must design around patents before the patented design achieves a wide install base to have a chance of introducing superior designs to the market. When patents on network effect technology are instead

³⁵ Eisenberg, *supra* note 16.

³⁶ *Id.* at 1076.

³⁷ *Madey v. Duke Univ.*, 307 F.3d 1351, 1362 (Fed. Cir. 2002) (exception only "for amusement, to satisfy idle curiosity, or for strictly philosophical inquiry").

³⁸ A network effect technology is one that becomes more valuable as more people use it. Computer software is a canonical example of a product with high network externalities, which arise from being able to exchange files among users, a larger support and training market, more add-on applications, etc. Kathleen Reavis Conner & Richard P. Rumelt, *Software Piracy: An Analysis of Protection Strategies*, 37 MGMT. SCI. 125, 136 (1991) (explaining why, in the presence of network externalities, software piracy can increase profits). The natural consequence of network externalities is monopoly, partially explaining the dominance of Microsoft Windows and Office on personal computers.

³⁹ It may of course be impossible to design around the patent. In such a case stagnation may be inevitable. As Bill Gates said when Windows 3.0 was new, "If people had understood how patents would be granted when most of today's ideas were invented, and had taken out patents, the industry would be at a complete standstill today." Bill Gates, *Challenges and Strategy* (May 16, 1991) (on file with the STAN. J.L. BUS. & FIN.).

⁴⁰ FED. TRADE COMM'N, *supra* note 29, ch. 2, at 22.

revealed late, competitors have no power to design around them. Instead they may resort to invalidating the patents, as played out in the aftermath of the Eolas patent verdict against Microsoft⁴¹ and a similar industry-wide effort which coalesced after a patent was claimed on part of the JPEG image file specification.⁴²

Because of network externalities, software competitors must have early knowledge of how broad the patent claims are. This is another reason why having widespread knowledge of patent claims is key for the economic incentive to design around. By requiring software patent owners to put competitors on notice of their patents, the economic benefits from designing around might not be thwarted.⁴³

3. Incentive to disclose

The disclosure theory recognizes that publicizing detailed technical knowledge is important to further research. It argues that some knowledge would stay secret without a patent system.⁴⁴ Society thus benefits in a second way from patents, in addition to the principal benefit of the inventor's commercialization of a new idea: the disclosure of the research so that others may build on it. Together they are more valuable, in a rose-colored world, than the patent monopoly's deadweight cost to society.

Economists have attacked this theory, arguing that patent disclosures are often too skimpy to be useful.⁴⁵ Eisenberg cites studies that argue "patent applicants often deliberately withhold important information from patent specifications so that they may continue to protect their 'know-how' through trade secrecy." One study found that a full one-half of all patents are so skimpy that they cannot be used to practice the invention on their own.⁴⁶ To be fair, such a determination must be on an individual industry basis. The computer software industry is famous for its uselessly

⁴¹ The World Wide Web Consortium joined with Microsoft and succeeded in having part of the patent overturned by the U.S. Patent and Trademark Office. Press Release, World Wide Web Consortium, World Wide Web Consortium Presents US Patent Office with Evidence Invalidating Eolas Patent (Oct. 29, 2003), available at <http://www.w3.org/2003/10/28-906-briefing>; *Microsoft Corp.: Rejection of Patent Is Hailed As Step Toward Resolving Suit*, WALL ST. J., Mar. 4, 2004, at B4 (patent office invalidated some claims of the Eolas patent).

⁴² Press Release, ISO SC29/WG1 Committee, Concerning Recent Patent Claims (July 19, 2002), available at <http://www.jpeg.org/newsrel1.html> (last visited Apr. 15, 2005).

⁴³ The marking proposal is set out *infra* in the text accompanying notes 202–05.

⁴⁴ Eisenberg, *supra* note 16, at 1028–30.

⁴⁵ See *id.* at n.52.

⁴⁶ Although these patents arguably should be invalidated for lack of enabling disclosure, it is not clear that the current legal standards would allow competitors to defeat the patents in court. See *infra* note 47. Moreover, given the extraordinary costs of patent litigation, see *infra* note 132, patent holders can be assured that most competitors will treat the patent as valid even when it is not. The patent holders thereby have almost all the advantages of a patent monopoly without disclosing trade secrets.

meager patent specifications.⁴⁷ Commentators have pointed out that software patents often disclose no more than what is apparent from using the software.⁴⁸ The same is true for simple mechanical inventions.⁴⁹ In these cases the patent disclosure is worth little.

For industries where most innovation occurs at the factory, such as chemical engineering, Judge Pauline Newman has observed that 85%-90% of technical information appears only in patent specifications.⁵⁰ There are also, notwithstanding the previous paragraph, some software patents that disclose meaningful technical information.⁵¹ In these situations the incentive to disclose provides real benefits.

The economic benefits bestowed by technical information in patent disclosures vary by industry. It is significantly less for computer software than some other kinds of technology, but it does not exist at all when researchers interested in the information cannot find it in the patent libraries or learn the patents through other means. Software makers might ensure that competing researchers know about relevant inventions by listing the patents that their various programs use.⁵²

4. Solution to Arrow's information paradox

Patents also allow for private information disclosure by solving Kenneth Arrow's information paradox, which describes the problem faced by an inventor selling an idea.⁵³ People, generally speaking, do not buy things sight unseen, so anybody contemplating the purchase of this idea will, naturally, want to know what it is. But once the inventor tells his idea to the prospective purchaser, he no longer has anything to sell. An erstwhile would-be buyer who walks away would know as much about the idea as somebody who actually completes the transaction. There is therefore an obvious opportunity for misappropriation and fraud. Because

⁴⁷ The Federal Circuit went so far as to issue a brace of decisions in 1997 holding that a software patent need not even use the words "software" or "computer," let alone disclose any details of the implementation. *Robotic Vision Sys., Inc. v. View Eng'g, Inc.*, 112 F.3d 1163, 1166 (Fed. Cir. 1997) ("plainly apparent" that a computer and software would be necessary); *In re Dossel*, 115 F.3d 942, 946-47 (Fed. Cir. 1997) (requirement under § 112(6) that applicant disclose how to "reconstruct" data met by statement: "Known algorithms can be used for this purpose."). See also FED. TRADE COMM'N, *supra* note 29, ch. 3, at 49; *infra* note 96.

⁴⁸ FED. TRADE COMM'N, *supra* note 29, ch. 3, at 49.

⁴⁹ See, e.g., U.S. Patent No. 6,367,353 (issued Apr. 9, 2002) (describing and claiming an improved corkscrew).

⁵⁰ Hon. Pauline Newman, Luncheon Speech to the ABA-IPL Section (July 21, 1994). She has clarified that her comment was specific to the chemical engineering industry. E-mail from Pauline Newman, Federal Circuit Judge, to the author (Oct. 4, 2004) (on file with the STAN. J.L. BUS. & FIN.).

⁵¹ See, e.g., U.S. Patent No. 5,040,217 (Aug. 13, 1991) (detailed discussion of part of MP3 audio compression, with FORTRAN program listing).

⁵² The marking proposal is set out *infra* in the text accompanying notes 202-205.

⁵³ Kenneth Arrow, *Economic Welfare and the Allocation of Resources for Invention*, in RATE AND DIRECTION OF INVENTIVE ACTIVITY: ECONOMIC AND SOCIAL FACTORS 609, 614-16 (Nat'l Bureau of Econ. Res. 1962).

misappropriation cases are difficult to litigate,⁵⁴ smart inventors are reluctant to reveal their ideas in the absence of patent protection.

An inventor who first obtains a patent can more freely disclose his invention during licensing negotiations, knowing he can sue for infringement.⁵⁵ Although patent litigation is always expensive, it is easier when the patent is written with the licensed product in mind.⁵⁶ Because this use of patents is strictly between two contracting parties, it is not necessary for the patents in question to be widely known.⁵⁷

5. Incentive to commercialize

Patents reduce risk for manufacturers, providing an incentive to commercialize.⁵⁸ Implementing any idea always requires some amount of capital. There is always the risk that a new product will fail,⁵⁹ in which case the investors lose money. Without upside potential, capitalists will not invest in new ideas. Patents improve the likelihood of an invention's success by preventing competitors from copying the idea and taking a share of the profits. Patents thereby help bring inventions to market.

It can be hard to swallow the distinction between the incentives to commercialize and invent without concrete examples. A recent one is the Bayh-Dole Act of 1980, which allowed universities to obtain patents on government-funded research. It was intended "to promote the commercialization and public availability of [government-funded] inventions."⁶⁰ Until then, even good ideas known to

⁵⁴ For example, in trade secret there are defenses of independent invention and reverse engineering, which makes proving misappropriation difficult from a purely evidentiary standpoint. *Chicago Lock Co. v. Fanberg*, 676 F.2d 400, 404 (9th Cir. 1982). Also, an inventor may not sue third parties once a trade secret becomes public. ROGER M. MILGRIM, *MILGRIM ON TRADE SECRETS*, 1 § 1.05[1] 1-197 n.2 (2004) (trade secret status lost once secrecy is).

⁵⁵ In a few on-sale or public-use bar patent cases, where the sale or use occurred due to alleged misappropriation, the unsympathetic judge told the inventor he should have filed for a patent before shopping his idea around. *See, e.g., Vanmoor v. Wal-Mart Stores, Inc.*, 201 F.3d 1363 (Fed. Cir. 2000); *Evans Cooling Sys., Inc. v. Gen. Motors Corp.*, 125 F.3d 1448 (Fed. Cir. 1997); *Lorenz v. Colgate-Palmolive-Peet Co.*, 167 F.2d 423 (3d Cir. 1948). One could point out that an inventor *must* disclose his invention if he first files for a patent, although in practical terms the disclosure in the patent may not be useful. *See infra* note 220 and accompanying text.

⁵⁶ Also, the inventor might be awarded attorney fees in a case of such blatant infringement, making litigation more practical. *See* 35 U.S.C. § 285 (2005) (court may award attorney fees in exceptional cases).

⁵⁷ Although a strengthened marking duty would not directly improve how Arrow's solution is applied, it would not hinder it either. *See infra* note 150 and text accompanying notes 210–211.

⁵⁸ Eisenberg, *supra* note 16, at 1036–44.

⁵⁹ It could also be an improvement on an older product.

⁶⁰ 35 U.S.C. § 200 (2005).

investors were not being commercialized because the risk of being undercut by competitors was too high. Judge Giles Rich also gave two historical examples.⁶¹

One might argue that competitors would only enter if they expected to make profits, and so the incumbent would continue to make profits even without a patent. Indeed, the market may be very profitable, and copycat competitors would not enter if it were not. The *expectation* of profit at the time the original manufacturer decides whether to enter is most important. The incumbent needs to expect to make a lot of money on some products to make up for losses on others—losses that the copycat manufacturers would not bear. Competitors may also seek to license the patent with the threat of litigation, which does not change the result that the original manufacturer loses expected profits because others enter.⁶² The original manufacturer cannot demand all of the industry profits, otherwise the competitors would not agree to the license agreement.

A patent, in a sense, encourages commercialization by sending a signal to competitors that a market has been claimed. To operate, the signal must be visible, and so it is important that patents be widely known among competitors in order to give a greater incentive to commercialize. The cheapest way to keep competitors out is to inform them of the patent rights and make a credible threat of legal action. If the competitors have yet to invest any money in the commercialization and the patent is watertight, the competitors would risk losing millions, compared to their present position, if they entered the market. On the other hand, if the competitors have already invested in capital equipment, the millions of dollars is a sunk cost and they risk little by suing to invalidate the patent.⁶³ The marking proposal in this article is a rather direct way of sending signals about which software is patented; all software

⁶¹ Judge Rich mentions a plastic made from sawdust invented by the government in the 1930s, and a chair for invalids. Giles S. Rich, *The Relation Between Patents Practices and the Anti-Monopoly Laws*, 24 J. PAT. & TRADEMARK OFF. SOC'Y 159, 178–79 (1942). In addition to writing the modern Patent Act in 1952, he is famous for his nonagenarian tenure on the Federal Circuit and its predecessor court.

⁶² There might also be lower total industry profit to allocate through licensing, also reducing the incumbent's profits. This is because splitting the production among several companies might lower total profit through reduced economies of scale, although the entrant's economies of scope might increase total profits if the entrant is a large, diversified company.

⁶³ To make the point more concretely with a simple model, let p be the likelihood of prevailing in litigation, $profit$ be the expected gross profits from the invention (not counting other costs listed here), $capital$ be the fixed costs (those that would be wasted if the product didn't come to market), and $lit-cost$ be the cost of litigation.

If a competitor learns about a patent before investing in a product, he will bring it to market only if he expects to make a net profit after litigation: $p \bullet profit - capital - lit-cost > 0$. If, on the other hand, he learns about the patent after investing in fixed equipment, he may be willing to litigate and enter the market as a way of cutting its losses, even if he no longer expects to turn a profit: $p \bullet profit - lit-cost > 0$.

would be marked with the manufacturer's patent numbers covering that technology.⁶⁴

One could argue that the incentive to commercialize is largely limited to government-funded research seeking private investors to bring the products to market, as the examples listed above display.⁶⁵ When private companies fund research, it is with an eye towards bringing the invention to market and thus there may be no need to provide an additional incentive to commercialize. Patents, moreover, are not the best lever for commercialization. Some commercially viable ideas are not patentable, like selling old products in smaller-sized packaging or other "obvious" ideas. A manufacturer might be unwilling to take a risk even if society would benefit from the new product.⁶⁶ Conversely, patents may give the manufacturer "too much" money, in the sense of creating deadweight loss. The incumbent may already benefit from a first-mover advantage to the market, or a lead-time advantage, or complementary effects with other products made by the manufacturer.⁶⁷

Yet there is another incentive to commercialize more relevant to industries already speckled with patents, when the risk of patent liability is greater than the risk of failure in the marketplace. It does not come from awarding patents to inventors like the incentives discussed above; rather it comes from having clear information about what technologies are patented by its competitors.⁶⁸ If a company knows about the most relevant patents in its industry, it can bring products to market without fear of treading on competitors' patents. Otherwise, technology may stagnate from hand-wringing even when improvements would be commercially attractive. One example of this incentive failure in the software industry, discussed later, is in the area of data compression.⁶⁹ If software-patent holders were required either to notify competitors of relevant patents or effectively to forfeit them, the

⁶⁴ The marking proposal is set out *infra* in the text accompanying notes 202–05. One might point out that if companies would truly benefit from signaling their commercialization efforts to competitors, they would already do so by marking their products. That software companies do not signal their competitors suggests either that the incentive to commercialize is small in practice, *see infra* text accompanying note 65, or at least that the benefits are small in the face of transaction costs and software's rapidly advancing technology.

⁶⁵ *See supra* text accompanying notes 60–61.

⁶⁶ *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 17–18 (1966) (commercial success, while a secondary indicium of non-obviousness, is not dispositive).

⁶⁷ Richard C. Levin, *A New Look at the Patent System*, 76 AM. ECON. REV., PAPERS AND PROCEEDINGS OF THE NINETY-EIGHTH ANNUAL MEETING OF THE AMERICAN ECONOMIC ASSOCIATION, May 1986, at 199–202. There are, in a sense, "too few and too many" patents in today's system, although it is unclear how one would formulate an improvement or whether it would be worth the transaction costs.

⁶⁸ *Nike, Inc. v. Wal-Mart Stores, Inc.*, 138 F.3d 1437, 1443 (Fed. Cir. 1998).

⁶⁹ *See, e.g., supra* note 96 and text accompanying notes 232–33.

competitors could be confident that they would not be penalized for treading on unknown patents.⁷⁰ There would be less disincentive to innovate.

II. Difficulty of searching software patents undermines economic justification

The economic justifications for software patents are undermined by the fact that there is presently no way to effectively search them. Software patents are not indexed in a meaningful way and keyword searches on the patents are ineffective. There are also, strangely enough, legal obstacles to searching the public domain patent libraries.

Software patents are so difficult to search, it is argued, that their purported economic benefits are undermined. All of the economic benefits depend to some degree on patent claims and technical disclosures being practically available to the public: invention (claims), designing around (claims and technical disclosures), disclosure (technical disclosure), and commercialization (claims). This problem must be solved if there is to be any pretense of an economic rationale for the software-patenting regime.

In part three it is shown how a strengthened marking requirement would address the obscurity problem and put software patents on the same footing as other kinds of technology.

A. Claims of software patents hard to classify

The U.S. patent collection, comprised of six to seven million granted patents, is sorted by the nature of the claims into 434 classes.⁷¹ The classes may be based on industry or use (e.g., butchering or farriery),⁷² proximate function (e.g., heat exchangers for liquids), effect or product (e.g., a complex shoemaking machine), or structure (e.g., the atomic structure of a chemical compound).⁷³ The patent office generally prefers to group inventions by proximate function because examiners need to search related fields to determine whether an invention is truly new. For instance

⁷⁰ A strengthened marking requirement, *see infra* text accompanying notes 202–05, would not protect against broad patents in other fields, *see infra* text accompanying notes 87–89 and 207, but it would still be an improvement over the status quo.

⁷¹ EDWARD EARLS ET AL., EXAMINER HANDBOOK TO THE U.S. PATENT CLASSIFICATION SYSTEM § I-B (2003). The number of patent classes varies constantly as the patent office tunes its classification scheme. Counting the utility classes listed on the patent office web site, disregarding the design and plant patent classes, gives the number 434. Patent and Trademark Office, *US Classes by Number with Title, at* <http://www.uspto.gov/go/classification/selectnumwithtitle.htm>.

⁷² They are class numbers 452 and 168, respectively. U.S. PATENT AND TRADEMARK OFFICE, U.S. DEP'T OF COMMERCE, MANUAL OF PATENT EXAMINATION PROCEDURE, § 903 (2004).

⁷³ EARLS, *supra* note 71, § I-A.

an examiner would want to know about heat exchangers for milk when he is deciding whether to grant a patent for a heat exchanger for beer.⁷⁴

These classes are further divided into about 150,000 subclasses, or “art units.” These art units overlap, so the patent office has devised an elaborate system for assigning patents to particular classes and subclasses.⁷⁵ The patent office has refined the system over many years to suit the needs of its patent examiners.⁷⁶ Inventions are allowed to be cross-referenced into multiple art units, which in theory ameliorates the problem of rigidly classifying inventions. The problem is that cross-references are limited by the examiner’s imagination at the time he makes the cross-references, in light of then-current technology. There is no reason to think that an examiner would stumble across an old patent and make the appropriate cross-references.

The system is not perfect for every need. There is room for error in an elaborate system of human judgment calls sorting patents into 150,000 subclasses. Finding a specific patent is difficult because it requires that the searcher have enough information about the claims and the same view of their significance as the original patent examiner. In addition, there are filing mistakes even at the coarser level of class instead of subclass.⁷⁷ Although patent examiners may not need to find every last patent related to a specific invention, potential infringers may want to conduct a thorough patent search to avoid ruinous risk to their businesses, which the classification system does not support. The classification system is not good for economic analysis of the patent system, either.⁷⁸ Finally, the computerized data are not perfect.⁷⁹

⁷⁴ *Id.*

⁷⁵ *Id.* at § IV.

⁷⁶ *Id.* at App. A.

⁷⁷ Two very similar insulating paper cup-holders were put under the classes “receptacles” (No. 220) and “envelopes, wrappers, and paperboard boxes” (No. 229). Perhaps the divergent classifications explain why the examiner for the later-issued patent did not find the earlier one, as evidenced by the lists of prior art printed on the patents. U.S. Patent No. 5,425,497 (issued June 20, 1995); U.S. Patent No. 5,205,473 (issued Apr. 27, 1993).

⁷⁸ Economic analysis is critical to informed policy-making about the patent system through legislation and regulations, rather than letting the law lurch from one court decision to the next. The complaint of economists is that the classification system ignores industry and product grouping.

The resulting classification system is based primarily on technological and functional principles and is only rarely related to economists’ notions of products or well-defined industries. ... A subclass dealing with the dispensing of liquids contains both a patent for a water pistol and for a holy water dispenser. Another subclass relating to the dispensing of solids contains patents on both manure spreaders and toothpaste tubes. Nevertheless ... almost all attempts to relate patent numbers to industrial data use the subclass system as their basic unit of assignment.

Software patents are particularly difficult to categorize because none of the four categorizing philosophies work very well. More specifically, algorithms are inappropriately indexed by specific use, and it is not practical to index broad patents. There are distinctions between software and other areas of technology that cause the categorization problems.

1. Indexed by specific use, not algorithm

The patent office has four philosophies for its classification system. Patents used to be sorted based on specific “industry or use,” but this philosophy has the serious drawback of separating “physically similar art . . . without a meaningful distinction.”⁸⁰ There is a like difficulty with sorting them by the end result.⁸¹ The modern preference is to put inventions together which work in a similar way.

Software inventions are, nevertheless, indexed not by the algorithm, but by the industrial use. Formerly the requirement was formal, whereas now it is enforced by the categorization system of classes and subclasses based on end results. The patent office classification scheme, therefore, is not helpful for finding patents claiming a particular algorithm.

Attorneys used to disguise patent applications as being directed towards specific industrial applications of the algorithms. The algorithms were not classified by the kind of algorithm but by the industrial application. They did this because courts formerly restricted software patents to specific industrial applications. Two remarkable cases in the late 1970s demonstrate how this obfuscation worked.⁸² These cases concerned the same technology and were decided unanimously by nearly identical five-judge panels; yet the outcomes could not be more different. The court in *In re Johnson* allowed a patent on a method of filtering noisy seismic records, but a “mere” mathematical algorithm to separate seismic signals was held to be not

Zvi Griliches, *Patent Statistics as Economic Indicators: A Survey*, 28 J. ECON. LITERATURE 1661, 1666 (1990). In fact, American economists sometimes use Canada’s patent libraries for empirical research, as they are more appropriately organized for that purpose.

What the economists want is, unfortunately, somewhat inconsistent with the goal of doing good prior art searches to prevent bad patents from issuing. See EARLS, *supra* note 71, § I-A (classifying by industry or use makes it more likely that two patents issue for the same invention). In that sense there is no such thing as a perfect patent classification system.

⁷⁹ For example, a patent for a method of “catalytic reduction of nitrogen oxides” has been cross-classified into “burial garments” and “trousers and overalls,” though the correct art units are printed on the patent itself. U.S. Patent No. 4,935,393 (issued June 19, 1990).

⁸⁰ EARLS, *supra* note 71, at § I-A. There is another approach not mentioned in the text to categorizing prior art (as a physical structure), which is most relevant to chemicals, not computer programs.

⁸¹ *Id.* This organization is by “effect or product,” a kind of “industrial or trade grouping.” It has largely the same drawbacks as sorting by “industry or use.”

⁸² *In re Johnson*, 589 F.2d 1070 (C.C.P.A. 1978); *In re Walter*, 618 F.2d 758 (C.C.P.A. 1980).

patentable in *In re Walter*.⁸³ The distinction was that the patent in *Johnson* was written towards a concrete application.

Today, software inventions need not be disguised as industrial machinery, yet they are still largely classified by their “effect or product.”⁸⁴ Perhaps the reason is the minimal disclosure requirements for software inventions. It is not possible to classify software inventions by their algorithms given that inventors are not required to describe detailed workings in the first place.⁸⁵ In any case, although the patent office has loosened its standards with respect to the patent applications it will accept, it still pigeonholes software patents as if they were written for specific industrial applications. The problem will not go away as the oldest software patents expire.⁸⁶

The main problem with classifying algorithms by their industrial application is that the claims may be broader than that specific application and the patent becomes very difficult for users in other industries to find.⁸⁷ For example, in the mid-1980s a company called Quantel introduced and patented a specially built, television-editing machine. The patents were naturally classified as “Image signal processing circuitry specific to television,”⁸⁸ but the claims were broad enough to encompass image-editing software on a personal computer. The patents covered various aspects of “painting” on a virtual “canvas,” such as the stylus used to direct the on-screen brush, the soft-edged brush, and the canvas.⁸⁹ Although Adobe defeated Quantel, it was quite difficult because most of the exculpatory evidence was decades old.

The natural question is what Adobe would have done if it had known about the Quantel patents earlier. Adobe could have licensed the patent cheaply, when less was at stake. Adobe also could have invalidated the patent while the evidence was still fresh.

⁸³ *Walter*, 618 F.2d at 770.

⁸⁴ See *infra* text accompanying note 94 (citing examples “error detection/correction and fault detection/recovery”). What kinds of errors are detected? Mechanical errors? Resource errors? User input errors? What is the appropriate response? Aborting? Compensating and continuing at reduced performance? Switching to a backup? Stopping a less important process? There are differences between, say, running out of paper and running out of memory.

⁸⁵ See *supra* note 47. This failure of the patent classification system to handle computer software was predicted by the patent office itself several decades ago. See *infra* note 101.

⁸⁶ Given the large number of software patents in existence, any solution would need to address existing patents rather than just prospective ones. The marking proposal at the end of this paper covers both old and new software patents.

⁸⁷ Another problem is that the software patents are harder to search for the examiners. As observed *supra* note 78, using categories based on “industrial use” or an “industrial grouping” may result in patents being erroneously granted.

⁸⁸ The cross-references were also inadequate because they were all art units within the original class.

⁸⁹ Richard L. Phillips, *Computer Graphics in Court: The Adobe/Quantel Case*, SIGGRAPH COMPUTER GRAPHICS NEWSL. (Aug. 1998), at <http://www.siggraph.org/publications/newsletter/v32n3/contributions/phillips.html>.

2. “Fundamental” software patents hard to classify

These days the patent office no longer requires software to be aimed towards a specific purpose,⁹⁰ but the problem of categorizing algorithms is no closer to being solved. Some patents broadly claim features common to most modern computer programs. These patents cannot be classified as “seismographs” or other specific industrial uses, so new subclasses have been created for these broad software patents. The difficulty lies in finding the patents once classified.

To find patents that might cover a specific product (or potential product) using the classification system, a researcher needs to list the various characteristics of the product that might be patented, and then look in the subclasses corresponding to those characteristics. The industrial use of the product is certainly one such characteristic. The researcher will probably need to compare his product against others on the market to look for innovations. Suppose the product is some kind of machine. If one of the machine’s features is that the case easily opens with a magnetic latch, he may want to look in “Closure fasteners (Magnetic)” to see if the same design has been patented.⁹¹ The researcher is less likely to notice features shared by many products already on the market because common features fail to stand out.

Computer user interfaces are an example of a complex product with innumerable features shared by products from various manufacturers. Most modern programs share a similar look and feel, one that has evolved through decades of industry experience because software is unlikely to sell if it does not resemble existing products. Because of the many features in common, it may be unclear which user interface elements could be patented. I.B.M., for one, patented the “progress bar,” the on-screen gauge showing the user how much longer he must wait.⁹² It is very easy to write an infringing progress bar, and it might never occur to a software developer that he should check such a trivial feature for patents. These six lines of code infringe the patent.⁹³

⁹⁰ See *infra* note 184.

⁹¹ If the machine were a laptop computer, he might find that it infringed Apple’s patent on the Powerbook and iBook latch. U.S. Patent No. 6,659,516 (issued Dec. 9, 2003).

⁹² U.S. Patent No. 5,301,348 (issued Apr. 5, 1994). Progress bars are widely used with software installation, document printing, file downloading, faulty disk correction, intensive computation, and any other computer function that involves significant delay.

I.B.M. is mentioned several times in this article, not out of enmity, but because its patent (and software patent) portfolio dwarfs those of all other companies, and it is widely perceived as the greatest threat to smaller companies. Robert P. Merges, *The Uninvited Guest: Patents on Wall Street*, ECON. REV. (FED. RESERVE BANK ATLANTA), Oct. 1, 2003.

⁹³ There are seven lines listed, but the first is a comment. Roland Stigge, *Analysis of EP0394160: Dynamic progress marking icon (IBM (US))*, (Dec. 28, 2002) at <http://www.rolandstigge.de/computer/patentviolation/EP0394160/description.txt>.

```
#!/bin/sh
echo -n $'_____ \r'
for ((i = 0; i < 10; i++)); do
echo -n "#"
sleep 1
done
echo
```

The progress bar patent also demonstrates how such “fundamental” patents are easily misclassified. It is buried in “Error detection/correction and fault detection/recovery,” a curious choice because progress bars have nothing to do with errors or faults. They merely tell the user how much longer to wait.⁹⁴

There is another way to look at the problem of categorizing broad software patents: A category should help narrow the number of patents to inspect individually. If a category applies to virtually all software, ipso facto it fails to narrow patent searches, and therefore is not a useful category. It is difficult for third parties to perform an infringement search of broad software patents using the classification system.

“Fundamental” patents should be distinguished from grossly overbroad patents, as in the Intertrust patent suit below. I.B.M. may have been the first company to im-plement a progress bar in graphical computer software, but as shown above it is very easy to provide a working implementation of the idea. There may have been no reason under Federal Circuit precedent to reject the patent application; “obvious” inventions are those suggested by prior art, not merely trivial ones. There are other fundamental software patents applying to most computer programs that are less trivial to implement. The problem, stated here, is that these everyday patents are difficult to find in the patent libraries.⁹⁵

3. How classifying software patents is different

Why are software patent claims broader, and therefore harder to classify, than other kinds of technology? Professors Burk and Lemley have argued that software patents tend to claim abstract ideas rather than concrete implementations.

[If a patent is] the first program to perform a given function [or, if for evidentiary limitations, the fact that it isn’t cannot be proven,] [it will be held to be non-obvious]. [Patents that] meet this test . . . will not be constrained by prior art to claim only their particular implementation of a function. They can claim the function itself. The fact that they give little or no description of how to

⁹⁴ The patent was eventually cross-referenced into a more useful category, but by then progress bars had become a standard part of computer user interfaces.

⁹⁵ The progress bar is a good example of a “harvested” patent. Part three discusses the costs of these patents and how companies would be dissuaded from patenting trivial software inventions if they had to keep track of which programs implemented them.

achieve this function will not bar the broad claims because the Federal Circuit has proven remarkably unwilling to require software patentees to disclose details. As a result, “we should expect the first programmer to implement a new idea in software to encompass the entire category of software, regardless of how second-comers actually implement the same concept.”⁹⁶

Patents are, as a general rule, required to claim no more than they disclose, and to disclose concrete implementations of the inventions. This requisite can be justified a few ways. Sometimes courts say that the disclosure teaches the public about the technology as a quid pro quo for the patent monopoly.⁹⁷ Other times it is described as something like an antifraud measure, to ensure that the inventor truly had invented what he claimed.⁹⁸

By implication patents are not to be given “for vague intimations of general ideas,”⁹⁹ but this is precisely the sad state of affairs concerning computer programs. Although the patent disclosure standards for software were once equally high,¹⁰⁰ they have eroded to the point where no implementation details need to be disclosed. In fact, it is uncommon to draft software patents based on nothing more technical than PowerPoint presentations, advance copies of user manuals, or two to three page, handwritten “invention disclosure statements.”¹⁰¹ These low disclosure standards for software patents mean that inventors can make very broad claims—far

⁹⁶ Burk & Lemley, *supra* note 25, at 1594.

⁹⁷ *Kewanee Oil Co. v. Bicron Corp.*, 416 U.S. 470, 484 (1974) (a patent specification’s technical disclosure is “the quid pro quo of the right to exclude”).

⁹⁸ *Gentry Gallery, Inc. v. Berkline Corp.*, 134 F.3d 1473, 1479 (Fed. Cir. 1998) (“the patent specification must clearly allow persons of ordinary skill in the art to recognize that the inventor invented what is claimed”).

⁹⁹ *Genentech, Inc. v. Novo Nordisk A/S*, 108 F.3d 1361, 1366 (Fed. Cir. 1997). *See also* *Rubber-Tip Pencil Co. v. Howard*, 87 U.S. 498, 507 (1874) (“An idea is itself not patentable, but a new device by which it may be made practically useful is”).

¹⁰⁰ *In re Doyle*, 482 F.2d 1385, 1390 (C.C.P.A. 1973) (describing even algorithms and flowcharts are “more in the form of indications of the results desired than any description of how the computers are operated to provide those results”); *In re Naquin*, 398 F.2d 863, 866 (C.C.P.A. 1968) (addition, multiplication are within the skill of an average programmer and need not be described in patent).

¹⁰¹ Conversely, the patent office often cites non-technical references, such as the basic computer tutorial books commonly sold at Wal•Mart, as prior art in its office actions. The gloomy prediction of the patent office in 1966 has come true.

The Patent Office now cannot examine applications for programs because of the lack of a classification technique and the requisite search files. Even if these were available, reliable searches would not be feasible or economic because of the tremendous volume of prior art being generated. Without this search, the patenting of programs would be tantamount to mere registration and the presumption of validity would be all but nonexistent.

Nelson Moskowitz, *The Metamorphosis of Software-Related Invention Patentability*, 3 COMPUTER/L.J. 273, 282 (1982) (quoting PRESIDENT’S COMMISSION ON THE PATENT SYSTEM, “TO PROMOTE THE PROGRESS OF . . . USEFUL ARTS” IN AN AGE OF EXPLODING TECHNOLOGY 13 (1966)).

broad than their own technical contributions to society—and expect their patents to be upheld.

The Quantel patent and I.B.M. progress bar patents are excellent examples of broad patents that lay unnoticed for years in an inappropriate category. Yet much broader software patents exist. The Intertrust portfolio of patents claimed virtually all ways of delivering movies, music, and other media over the Internet. Not surprisingly, the software failed to ship for years after the first patent was filed, and what customers eventually received fell short of expectations.¹⁰² Despite the patent specification's lack of supporting detail, the industry consensus was that Intertrust's patents would probably be upheld in court.¹⁰³ Microsoft paid half a billion dollars in a settlement, suggesting that it had faced a very real possibility of a verdict in the billions of dollars against a company that never had a viable product.¹⁰⁴ It is even possible that Microsoft would have been driven out of business, as the Intertrust patents cut across most of Microsoft's product line.¹⁰⁵ In the early 1990s another computer company stood its ground against a patent believed by some observers to be without merit.¹⁰⁶ It lost, received a permanent injunction, and then liquidated.¹⁰⁷

Patents can effectively bar use of even decades-old algorithms. For example, arithmetic coding was invented more than four decades ago, but due to intensive

¹⁰² Roger Parloff, *Can Victor Shear Bring Down Microsoft?*, FORTUNE, Dec. 17, 2002, available at <http://news.dmusic.com/article/5853>. The Intertrust patent specification was a thousand pages long, yet the implementation details that would have made the Commerce Architecture a viable product weren't there.

¹⁰³ One might ask why competitors don't simply defeat "bad" software patents in court. The answer is that the enablement standard for software patents is so low that it may not be possible to defeat them. Microsoft would not have paid half a billion dollars to settle the case if it thought it could win. The risk of losing also might be unacceptably high.

¹⁰⁴ Kodak was hit with the largest patent infringement verdict in history in 1990 when it was ordered to pay \$909.5m to Polaroid. Although that verdict was large, on face suggesting that Microsoft's settlement was reasonable, half of the Kodak verdict was interest owing to the 14-year duration of the trial, and the rest reflected lost sales by Polaroid. Lawrence Ingrassia & James S. Hirsch, *Polaroid's Patent-Case Award, Smaller Than Anticipated, Is a Relief for Kodak*, WALL ST. J., Oct. 15, 1990, at A3. Intertrust, on the other hand, had no lost sales, as it never had a viable product.

¹⁰⁵ Microsoft's founder Bill Gates presciently observed in 1991,

I feel certain that some large company will patent some obvious thing related to interface, object orientation, algorithm, application extension or other crucial technique. If we assume this company has no need of any of our patents then the[y] have a 17-year right to take as much of our profits as they want.

See supra note 39 (typographical error in original).

¹⁰⁶ Richard Stallman & Simon Garfinkle, *Viewpoint: Against Software Patents*, COMMUNICATIONS OF THE ACM, Jan. 1992, at 18 (invention trivial and in use at least five years before patent filed).

¹⁰⁷ The company was Commodore Computer. KEVIN G. RIVETTE & DAVID KLINE, REMBRANDTS IN THE ATTIC: UNLOCKING THE HIDDEN VALUE OF PATENTS 121 (2000).

research there are a number of patents in the area.¹⁰⁸ The existence of so many patents has prevented the use of other variants of arithmetic encoding.¹⁰⁹

B. Lack of good keywords for software patents

The other way to search the patent library is by keyword. If one is interested in patents pertaining to specific processes,¹¹⁰ components,¹¹¹ or the like, a keyword search can thresh out most of the chaff. Keyword searches work best when the terms are standardized. Unfortunately for browsers of the patent libraries, inventors may supply their own keywords and have considerable latitude in doing so. They may even go so far as to redefine their own words, so long as a term isn't "given a meaning repugnant to [its] usual meaning."¹¹²

The tendency to abuse words seems to be checked in more established fields like mechanical engineering, perhaps because in most fields, inventors are unable to gain additional patent breadth by using vague or non-standard terminology; there are no synonyms for "gold." As discussed previously, in more traditional fields inventors are required to make detailed disclosures to support all of the claims they make. Using vague terminology does not expand the scope of the claims. To the contrary, an indistinct specification will be unable to support as many claims in a traditional area of engineering because it lacks essential detail.

Computer science claims, on the other hand, often refer to processing "data" or "documents,"¹¹³ words that could be read to comprise any kind of digital information whatsoever. A search on "data" could not narrow a patent search in a

¹⁰⁸ Arithmetic coding is a form of entropy compression giving results about 5% better than Huffman coding. It was first described in 1963, based on a previously unpublished manuscript by Elias. NORMAN ABRAMSON, INFORMATION THEORY AND CODING 61–62 (1963). Decades later, I.B.M. was issued several patents concerning arithmetic coding. See *infra* note 221. Patents are also held by Mitsubishi and A.T.&T. ROY HOFFMAN, DATA COMPRESSION IN DIGITAL SYSTEMS 48 (1997).

¹⁰⁹ For example, the *bzip* compression program was terminated in favor of *bzip2*, which used Huffman coding in the back-end. Although *bzip* did not use IBM's Q-coder and was believed to be outside the scope of the patents, Julian Seward, the author of the programs, thought the risk of infringement was too great for *bzip* to become widely used. Posting from Paul Sootman, paul@wau.mis.ah.nl, to debian-devel@lists.debian.org (Dec. 15, 1997) (on file with STAN. J.L. BUS. & FIN.).

¹¹⁰ Searching for "chemical vapor deposition" turns up 50,778 patents, less than one percent of the entire corpus, although the text of patents before 1975, amounting to slightly over half the patent corpus, is not indexed by the patent office.

¹¹¹ "Platinum" finds 100,408 patents, about one and a half percent of issued patents.

¹¹² U.S. PATENT AND TRADEMARK OFFICE, U.S. DEP'T OF COMMERCE, MANUAL OF PATENT EXAMINING PROCEDURE § 608.01(o) (2004). The abstract may be a more fruitful target of keyword searches; it is required to be an "adequate and clear" statement of the contents of the disclosure, allowing the reader "ascertain quickly the character of the subject nature covered by the technical disclosure." *Id.* at § 608.01(b).

¹¹³ See, e.g., U.S. Patent No. 5,414,810 (issued May 9, 1995) (claiming two windows containing data, where the second appears only once when the first is clicked); U.S. Patent No. 6,820,236 (issued Nov. 16, 2004) (claiming pre-fetching linked documents over a network).

meaningful way. Conversely, a search on a particular kind of data will miss patents drafted in terms of “data.”

Inventors may need to make up new words to describe the software they have written because computer science is an emerging field.¹¹⁴ If these made-up words fail to catch the attention of the industry at large, the patents written with the words become practically invisible for researchers using standard search terms. One overbroad¹¹⁵ AT&T patent claimed what is now known as “backing store for windows,”¹¹⁶ but the claims were written instead with the idiosyncratic word “layers.”¹¹⁷ The key phrase “backing store” appeared nowhere in the patent. Peculiar jargon is probably more likely to find itself in patents because inventors, by definition, believe they have cutting-edge work and may think that new words are necessary to properly describe it.

Other times terminology may change simply due to the rapid progress of the computer industry. The six leading makers of spreadsheet software were sued in 1990 with a patent that didn’t use the word “spreadsheet” anywhere, doubtlessly because the patent was so old that at the time it was written, the word “spreadsheet” still referred to a leaf of ledger paper.¹¹⁸ The case is taken up in more detail later in this Article.¹¹⁹

¹¹⁴ The same problem occurs in other emerging fields, such as nanotechnology. Barnaby J. Feder, *Tiny Ideas Coming of Age*, N.Y. TIMES, Oct. 24, 2004, at Week in Review (referring to patent office “confusion in a realm where the same invention might be called a carbon nanotube, an elongated cylinder made of carbon or a carbonaceous cylinder in three separate patent applications”).

¹¹⁵ The patent threatened to stop M.I.T. from giving technology it invented to the public. Stallman & Garfinkle, *supra* note 106, at 19. The X Window System software actually used a simpler, older backing store algorithm by M.I.T. Richard Stallman & Simon Garfinkle, *Response to Dennis Ritchie*, COMMUNICATIONS OF THE ACM, June 1992, at 15. Unfortunately the older implementation had been deemed “too obvious to publish.” Dennis M. Ritchie, *Letter*, COMMUNICATIONS OF THE ACM, June 1992, at 13. Not being publicly available, M.I.T.’s earlier work could not invalidate the overbroad AT&T patent.

¹¹⁶ U.S. Patent No. 4,555,775 (issued Nov. 26, 1985) was the subject of extensive legal maneuvering in the 1980s. AT&T sent infringement notices to members of M.I.T.’s X Consortium. John Markoff, *Patent Action on Software by A.T.&T.*, N.Y. TIMES, Feb. 26, 1991, at D1. The letter acknowledged the term “backing store.”

Consequently, we bring to your attention an AT&T patent #4,555,775 invented by Robert C. Pike and issued on November 26, 1985. The “backing store” functionality available in the X Windows System is an implementation of this patented invention, therefore, your company/institution needs a license from AT&T for the use of this patent. Letter from A.E. Herron, Manager, Intellectual Property, AT&T, to X Consortium Members, (Feb. 7, 1991) (copy on file with the STAN. J.L. BUS. & FIN.).

¹¹⁷ Rob Pike, *Graphics in Overlapping Bitmap Layers*, 2 ACM TRANSACTIONS ON GRAPHICS 135 (1983) (word “layers” coined by inventor Pike). Pike’s invention was hailed as a clever way of saving memory. JAMES D. FOLEY ET AL., *COMPUTER GRAPHICS: PRINCIPLES AND PRACTICE* 996–98 (2d ed. 1996).

¹¹⁸ See *infra* note 127 and accompanying text. The spreadsheet patent also neglected to say that the invention was effectively just a grossly inefficient topological sort. This defect,

Another cause of idiosyncratic patents is company and law firm culture. As John Allison said,

My study of many thousands of individual patents . . . led me to conclude quite some time ago that shortcuts don't work for finding software patents because they are frequently difficult to identify as software patents even when you have them in front of you on the computer screen. They are also highly idiosyncratic to the firms that own them—some of the language is often different among different firms . . . even in the same technology areas.¹²⁰

Patent examiners can do little to rein in software inventors' verbal creativity. Ideally in the course of a prior art search, the examiner would turn up relevant outside references using the correct terminology and pressure the applicant to use standard language. However, the prior art searches for software patent applications are amazingly incomplete. One study found that 80% of software patents are granted without *any* non-patent references,¹²¹ and it's not uncommon for the 20% of non-patent references to be aftermarket user manuals rather than detailed technical descriptions of how existing software actually works.

C. Infringement check is too expensive to be practical

There are several reasons that someone may want to know whether a given product infringes third-party patents. A potential competitor may want to know if she would face liability for introducing a new product. A company with a product on the market may want to know if any recently issued patents read on the product. A consortium may want to ensure that a new standard may be freely implemented.

Such a person must get a list of the patents that the product might infringe¹²² and check each patent carefully for possible infringement. This checking can be quite expensive.¹²³ Yet given the unpredictable nature of patent litigation, it may be impossible at any cost to know for sure whether a product infringes any patents.

however, was present from the time the patent was filed, rather than reflecting a change in the technical vernacular.

¹¹⁹ See *infra* text accompanying notes 126–130.

¹²⁰ E-mail from John Allison, Professor, University of Texas at Austin, to the author (Jan. 24, 2005) (on file with the STAN. J.L. BUS. & FIN.).

¹²¹ James Gleick, *Patently Absurd*, N.Y. TIMES MAG., Mar. 12, 2000, at 47.

¹²² The assumption here is that the list of potentially infringing software patents would be long and filled with false positives, as would be the case if the patent libraries were searched by keyword or category. If patent owners were required to state the patents covering their various computer programs, makers of competing programs would have a short and relatively accurate list to check.

¹²³ The cost of checking whether a particular product infringes a particular patent is just as high for the patent owner himself. See *infra* note 203.

The scope of a patent depends primarily on its claims, rather than the title of the patent or the abstract, and only in small part on the detailed description.¹²⁴ When checking a list of patents for possible infringement, it is therefore insufficient to scan the list for the most promising candidates and disregard the rest. It is entirely possible that the title and abstract will be vague or misleading, despite patent office policy intending the contrary.¹²⁵ The spreadsheet patent mentioned earlier is a prime example of a patent where the scope cannot be determined from the title, abstract, or detailed description.

The six largest spreadsheet vendors were sued in 1990 under U.S. Patent No. 4,398,249 (issued Aug. 9, 1983), which was filed twenty years earlier. Five of the vendors settled. The last won on a technicality after five years of litigation.¹²⁶ The product described by the patent was not a spreadsheet program; in fact, electronic spreadsheets would not be invented for another decade.¹²⁷ The patent instead described a mainframe programming language that made lists of formulas into standalone business programs for use by accountants.¹²⁸ These business programs were not spreadsheets, either; they actually did not use a computer screen at all. The user sat at a teletypewriter, with computer output streaming out on paper.¹²⁹ What this ancient mainframe program had in common with spreadsheets was that they both used a list of formulas that could be entered in whichever order the user found most convenient. The abstract and detailed description of Patent No. 4,398,249 might easily mislead an engineer who read them into thinking that the invention related only to programming languages and compilers.¹³⁰ The title did not mention the formula sorting feature at all.

¹²⁴ Claims are read “in view of” the patent specification. *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc), *aff’d*, 517 U.S. 370 (1996). There’s no guarantee that the title or the abstract wouldn’t be used to interpret the claims, but it isn’t a regular part of claim construction.

¹²⁵ U.S. PATENT AND TRADEMARK OFFICE, U.S. DEP’T OF COMMERCE, MANUAL OF PATENT EXAMINING PROCEDURE § 608.01(o) (2004).

¹²⁶ *REFAC Int’l, Ltd. v. Lotus Dev. Corp.*, 887 F. Supp. 539 (S.D.N.Y. 1995), *aff’d*, 81 F.3d 1576 (Fed. Cir. 1996).

¹²⁷ The metaphor of an interactive, electronic ledger wasn’t invented until a decade later by a different man. Dan Bricklin, *Was VisiCalc the “First” Spreadsheet?* (2004) at <http://www.bricklin.com/firstspreadsheetquestion.htm> (last visited Apr. 15, 2005) (Bricklin and Bob Frankston were the sole authors of VisiCalc). Messrs. Bricklin and Frankston knew nothing of LANPAR. John Kasdan, *Obviousness and New Technologies*, at 7 (Center for Law and Economic Studies, Columbia Law School, Working Paper No. 146, 1999).

¹²⁸ *Patent Protection for Software-Related Inventions: Hearing Before the Patent and Trademark Office, Dep’t of Commerce* (1994) (statement of Paul Heckel, Abraham Lincoln Patent Holders Ass’n).

¹²⁹ *Id.*

¹³⁰ Moreover, the description in the LANPAR abstract failed to describe what it was doing using the relevant term of art (a “topological sort”) and used an obtuse, grossly inefficient implementation.

Being forced to check the claims of a patent for infringement can cost \$20,000 to \$100,000 for a competent opinion of counsel on a single patent.¹³¹ Even at this cost, an opinion of non-infringement does not guarantee freedom from legal troubles. Defending against even a weak case of infringement can cost millions in legal fees and risk a costlier verdict.¹³² At these costs, plainly it is not possible for someone to thoroughly look for infringed patents.

Perhaps the best evidence that infringement is practically impossible to determine is that even patent holders do not necessarily know the scope of their claims. Large portfolio-holders like I.B.M.¹³³ and Texas Instruments¹³⁴ make billions of dollars by licensing their patents to smaller companies. They do so, not by determining the scope of infringement, but by blanket licensing most of their entire portfolio in exchange for a fixed percentage of gross sales or for a large fixed fee.¹³⁵ It is simply not economical to determine infringement precisely.

Even companies with only a few patents may not know their scope. A small company named Forgent did not realize until 2002 that one of its patents allegedly claimed one of the steps in producing JPEG image files.¹³⁶ In fact, it is not even clear whether the Forgent patent applies to JPEG.¹³⁷ In light of the abstruse way in which algorithms are claimed in software patents, it is not completely surprising that a company holding only forty patents would fail to uncover this claim for fifteen

¹³¹ Brief of Amicus Curiae Brief of Securities Industry Association in Support of Defendants-Appellants at 9, *Knorr-Bremse Systeme Fuer Nutzfahrzeuge GmbH v. Dana Corp.*, 383 F.3d 1337, 1347 (Fed. Cir. 2004) (Nos. 01-1357, 01-1376, 02-1221, 02-1256).

¹³² See Am. Intell. Prop. Law Ass'n (AIPLA), *2003 Report of the Economic Survey* 22, 93–94 (median cost of litigating a small single-patent case through trial in which \$1m–\$25m is at risk is \$2m; for trials where more than \$25m is at stake, the median cost is \$4m).

¹³³ I.B.M. earned almost \$900m in 2002 on “Sales and other transfers of intellectual property” and “Licensing/royalty-based fees.” *International Business Machines Corp., S.E.C. 10-K Filing for Fiscal Year Ended Dec. 31, 2003*, Exhibit 13, at 54 (filed Mar. 8, 2004).

¹³⁴ Texas Instruments earned about \$350m in 2002 on semiconductor royalties; its total patent income has been estimated to reach \$800m a year. *Texas Instruments, Inc., S.E.C. 10-K Filing for Fiscal Year Ended Dec. 31, 2003*, Exhibit 13, at 57 (filed Feb. 27, 2004); RIVETTE & KLINE, *supra* note 107, at 125. It was in fact Texas Instruments’ aggressive royalty-seeking which triggered a rash of semiconductor patent licensing. Bronwyn H. Hall & Rosemarie Ham Ziedonis, *The Patent Paradox Revisited: An Empirical Study of Patenting in the U.S. Semiconductor Industry, 1979–1995*, 32 RAND J. ECON. 101, 109–10 (2001).

¹³⁵ See *infra* notes 212–17 and accompanying text. One can argue that blanket licenses provide a benefit of reduced transaction costs, as in A.S.C.A.P. or B.M.I. music licensing. But that is a false analogy, because patent blanket licenses may confer no benefit. See *infra* text accompanying note 220.

¹³⁶ Amit Asaravala, *Forgent Sues Over JPEG Patent*, WIRED NEWS (Apr. 24, 2004), at <http://www.wired.com/news/business/0,1367,63200,00.html>.

¹³⁷ “The patent describes an encoding method that is clearly not like what JPEG does. The patent describes a three-way symbol classification; the closest analog in JPEG is a two-way classification. If the jury can count higher than two, the case will fail.” Gillian Law, *Forgent Claims JPEG Patent; Others Cry Foul*, NETWORK WORLD FUSION (July 19, 2002) (quoting Tom Lane, organizer of the Independent JPEG Group), available at <http://www.nwfusion.com/news/2002/0719jpeg.html>.

years.¹³⁸ Yet Forgent has collected at least \$100 million in royalties on the JPEG claim from companies unwilling to risk litigation.¹³⁹

D. Willful infringement: Legal obstacle to searching

It may seem strange that a patent system premised on dissemination of technical information would put up legal obstacles to searching these public records, but such is the law with “willful infringement.” This doctrine puts a very high burden on those who are aware of their potential infringement. In practical terms, this burden means that engineers cannot read patents in which they may be interested. If an infringer is found to have acted “willfully,” he can be held liable for triple the usual amount of damages for infringement, as well as the other side’s attorney fees.¹⁴⁰ To avoid a finding of willfulness, one who is aware of potential infringement must “exercise due care not to infringe.”¹⁴¹ Normally this means paying a lawyer tens of thousands of dollars for a favorable opinion letter and then continuing business as usual.¹⁴²

Willfulness penalties may not affect infringement, but they do affect the spread of patent information among non-lawyers. Engineers may want to read competitors’ patents to get a lay of the land.¹⁴³ They may innovate more with better knowledge of the state of the art and may be willing to create new designs to avoid existing patents.¹⁴⁴ However, in reading patents with useful information, they may encounter some that are irrelevant and some that their own product already infringes. If they are later sued on one of the infringing patents, the fact that they already knew of the potential infringement could result in a finding of willfulness and triple damages, unless they inoculated themselves by paying an attorney for a favorable opinion letter. There is therefore a cost imposed on engineers for reading

¹³⁸ See U.S. Patent No. 4,698,672 (issued Oct. 6, 1987); Mark Hachman, *Update: Forgent Claims Rights to JPEG Patent*, EXTREMETECH (July 18, 2002), available at <http://www.extremetech.com/article2/0,3973,389261,00.asp>.

¹³⁹ Dawn Kawamoto, *Graphics Patent Suit Fires Back at Microsoft*, CNET NEWS.COM (Apr. 22, 2005), at http://news.com.com/2100-1025_3-5681112.html.

¹⁴⁰ William F. Lee & Lawrence P. Cogswell, III, *Understanding and Addressing the Unfair Dilemma Created by the Doctrine of Willful Patent Infringement*, 41 HOUS. L. REV. 393, 396 (2004).

¹⁴¹ *Id.* at 407.

¹⁴² Favorable opinions offered at trial effectively insulate clients from enhanced damages. Imron T. Aly, *Encouraging Unprofessionalism: The Magic Wand of the Patent Infringement Opinion*, 12 GEO. J. LEGAL ETHICS 593, 597 (1999). Effectively the only constraint is that the letter “not be merely conclusory and include in it several expected components.” *Id.* at 610. Letters to be used at trial are written as subjective opinions, “in order to bolster the goals that [the client] had previously decided for itself.” *Id.* at 609, 605. See *supra* text accompanying note 131 (cost of opinion letters). Conversely there have only been a handful of cases where infringers avoided a finding of willfulness without an attorney’s letter to protect them. Lee & Cogswell, *supra* note 139, at 408 n.72.

¹⁴³ It is very easy to look for patents assigned to a particular company.

¹⁴⁴ Lee & Cogswell, *supra* note 139, at 448. See *supra* text accompanying notes 28–43.

patents, potentially a very great one if the colorable claims for infringement are numerous.¹⁴⁵

It may be worse if an engineer reads a patent and incorrectly decides that the technology is so different that an opinion letter is unnecessary,¹⁴⁶ because then the company is exposed to triple-damages liability without warning. It is quite possible for a non-lawyer to misapprehend the scope of a patent, particularly if the non-lawyer looks at the title, abstract, or detailed description instead of the claims.¹⁴⁷ Many engineers feel that the most prudent approach to avoid a finding of willfulness is to not read their competitors' patents, period.¹⁴⁸

The doctrine of willfulness is not going away anytime soon. Willfulness was not devised by Congress after weighing the costs. It was made from whole cloth by the patent judges on the Federal Circuit,¹⁴⁹ which chose to keep the doctrine largely intact after an en banc reexamination in late 2004.¹⁵⁰ The most viable alternative, if engineers are effectively prohibited from searching the patent libraries themselves, may be to have patent holders directly give notice to competitors' software engineers of the patent numbers most relevant to the patent holders' products.

E. Obscurity undermines economic incentives

The preceding discussion argues that software patents are difficult to find by category or keyword, impractical to check once found, and unwise to check in any event due to legal liability. The practical sum effect is that software patents are unavailable to be read by subsequent researchers.

The economic incentives, again, are those of invention, designing around, disclosure, and commercialization.¹⁵¹ All depend to some degree on patent claims and disclosures being publicly available. The difficulty of finding software patents through the libraries means that the multibillion-dollar costs of the software-patenting regime are unlikely to be offset.

The incentive to invent argues that researchers are more willing to flesh out ideas and experiment when there is potential to make a lot of money on the invention. In addition, society benefits from encouraging research. 3M's famous

¹⁴⁵ Lee & Cogswell, *supra* note 139, at 448.

¹⁴⁶ *Id.*

¹⁴⁷ See *supra* text accompanying notes 125–138.

¹⁴⁸ The founder of the Linux project, Linus Torvalds, see *infra* text accompanying notes 237–44, said as much. Gary Rivlin, *Leader of the Free World*, WIRED, Nov. 2003, at http://www.wired.com/wired/archive/11.11/linus_pr.html (last visited Apr. 15, 2005).

¹⁴⁹ Mark A. Lemley & Ragesh K. Tangri, *Ending Patent Law's Willfulness Game*, 18 BERKELEY TECH. L.J. 1085, 1109 (2003).

¹⁵⁰ *Knorr-Bremse Systeme Fuer Nutzfahrzeuge GmbH v. Dana Corp.*, 383 F.3d 1337, 1347 (Fed. Cir. 2004).

¹⁵¹ Patents also apply a solution to Kenneth Arrow's information paradox. As a mechanism for private contracting, it does not depend on public knowledge of patents. The first part of the paper discusses the incentives in detail.

chemical engineering efforts, for example, produced both major successes and accidental ones, like the Post-It note. It seems, though, that patents do not encourage computer software research in the same way as they do other kinds of technology. Software inventors are not encouraged by the potential for patents. They simply do not know whether their ideas are already covered by older, overbroad patents. They may try to look in the patent libraries for their idea, to find assurances that a patent would be worth pursuing. Yet if they look, the inventors will almost certainly find nothing. They must therefore have another way of learning about relevant patents.

The incentive to design around challenges competitors to find ways around existing patents. There may be net economic benefits from better, new designs. A new drug may have fewer side effects than one already on the market, for example. To find substitutes for patented inventions, entrants to the market need to read the patents to know where their search should begin. Finding the patents may be difficult for software engineers, as looking in the patent libraries may well be fruitless. Or worse, the search may produce only a subset of the relevant patents, resulting in wasting research and development efforts when the competitor is sued. The motivation to design around patents thus falls apart in the context of computer software, unless particular patents happen to be well-known for some other reason.

The disclosure of software patents theoretically could be very useful. In some areas of technology, like chemical engineering, most technical information is disclosed through patents. Although the technical disclosures of software patents are generally less informative, some are replete with information. Regardless of how much information the software patents provide, however, none of it is useful if the patents cannot be found. If the inventor used neologisms to describe the key parts of his idea, or if the patent examiner did not know how to categorize it, future researchers would probably not be able to benefit from the disclosure, even though patents are often described as a quintessential quid-pro-quo, requiring disclosure for a grant of monopoly. This is also true if future researchers are inhibited by the willfulness doctrine from reading patents in their area of interest.

The incentive to commercialize takes two forms with software patents. An entrepreneur may want to signal to competitors that a particular idea has been taken. Alternatively, an entrepreneur may want assurances that he will not be sued for using an idea that he thinks is in the public domain. Either way, for the patents to function as a signal, the relevant patents must be well-known in the industry. Relying on the patent libraries for finding patents falls short, particularly when there are legal obstacles against engineers searching the patent libraries!

In practice, most software patents are so obscure that they can hardly be thought of as public documents. The conclusion is that software patents do not support innovation, but perhaps software patents could be more aligned with economic principles if notice of relevant software patents were given to competitors

and the public. The third part of this paper describes an alternative mechanism—a strengthened, uniform duty to mark—to do just that.

F. Software patents arrive without economic justification

One might ask whether Congress considered the problems of software patent obscurity when it decided to allow patents on computer software. However, it was the courts, and not the legislature, that allowed software patents, and their justification relied little on informed policy-making. Courts instead allowed software patents because deciding patentability in individual cases had become troublesome.

Early Supreme Court cases disapproved of software patents. A long-standing rule of patent law is that mathematical formulae by themselves are not patentable.¹⁵² The principle was applied in *Gottschalk v. Benson* to an algorithm that could generally be calculated by hand, but had “no substantial practical application except in connection with a digital computer, which means that . . . the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself.”¹⁵³ The court invalidated another patent on an algorithm by the same reasoning in a subsequent case, *Parker v. Flook*.¹⁵⁴

The Supreme Court limited *Benson* and *Flook* a few years later in *Diamond v. Diehr*,¹⁵⁵ when it allowed a patent on a computer-controlled rubber mold, reasoning that curing rubber is an industrial process and patents were intended to protect industrial processes.

Over the next two decades, the Federal Circuit used the toehold of *Diehr* to allow patents on all “useful” software.¹⁵⁶ It first interpreted *Diehr* to allow patenting on inventions producing a “useful, concrete, and tangible result,”¹⁵⁷ which then was held to encompass financial “numbers”¹⁵⁸—the legerdemain being that one does not normally think of a “number” as being “able to be seen or touched because it exists in reality, not just as an idea” and “able to be touched or perceived through the sense

¹⁵² “While a scientific truth, or the mathematical expression of it, is not patentable invention, a novel and useful structure created with the aid of knowledge of scientific truth may be.” *McKay Radio & Telegraph Co. v. Radio Corp. of America*, 306 U.S. 86, 94 (1939).

¹⁵³ 409 U.S. 63, 71-72 (1972).

¹⁵⁴ 437 U.S. 584 (1978).

¹⁵⁵ 450 U.S. 175 (1981).

¹⁵⁶ The Federal Circuit is a centralized court set up to handle, among a handful of other things, all patent appeals from district courts and the patent office. 28 U.S.C. § 1295 (2005). It is the United States’ thirteenth court of appeals, after the D.C. Circuit and the eleven numbered circuits, but its jurisdiction is based on subject matter rather than geography.

¹⁵⁷ *In re Alappat*, 33 F.3d 1526, 1544 (Fed.Cir. 1994) (en banc). The “useful, concrete, and tangible result” in *Alappat* was a smooth line on an oscilloscope display.

¹⁵⁸ The court said tangible, concrete numbers included “price, profit, percentage, cost, or loss.” *State Street Bank & Trust Co. v. Signature Fin. Group*, 149 F.3d 1368, 1375 (Fed. Cir. 1998).

of touch,” which are the relevant definitions of “concrete” and “tangible,” respectively.¹⁵⁹

Then a year later the court dispensed with any limitation on patenting “useful” computer software.¹⁶⁰ Without comment from Congress, software patentability has become “a matter for the history books.”¹⁶¹

III. Solving the notice problem with a uniform marking duty

Because the patent libraries are so difficult to search, Congress long ago passed a law obligating patent owners to mark their goods with patent numbers.¹⁶² If they fail to do so, they are barred from collecting any money from infringers. This duty is why many products for sale in the U.S. have patent numbers printed on them.

The problem is that this marking duty does not suit computer software very well. First, it is largely ignored, for limitations in the law explained later. Another problem is that due to strong network effects underlying computer software, the present marking duty’s penalty of temporarily disallowing monetary damages usually has little effect on patent owners’ decision whether to mark patented products. A patent owner would still have the devastating remedies available of injunctions and future royalties.

There are swarms of software patents in effect, virtually impossible to search and not publicized through marking, with consequent negative economic effects. Therefore, applying the marking duty uniformly to computer software would lessen the practical difficulties.

A. The present marking duty

Congress recognized long ago that patent records are practically impossible to search, and thus enacted a marking duty so that patent owners would shoulder some of the burden of letting the public know which patents cover which ideas. Competitors can accordingly see which patents cover a product merely by inspecting the merchandise.

1. Marking required for damages

¹⁵⁹ ENCARTA WORLD ENGLISH DICTIONARY (1999).

¹⁶⁰ AT&T Corp. v. Excel Communications, Inc., 172 F.3d 1352, 1359 (Fed. Cir. 1999).

¹⁶¹ Julie E. Cohen & Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 CAL. L. REV. 1, 4 (2001).

¹⁶² The statute, 35 U.S.C. § 287 (2005), says patent holders “may” mark, but in light of the consequences for failure to do so it is usually described as a duty. *See, e.g.*, DONALD S. CHISUM, CHISUM ON PATENTS, § 20.03[7][c] (“The Patent Act imposes a duty to mark on patent owners.”).

A long time ago, learning which patents covered which products required a horseback trip to Washington, D.C. to visit the patent office.¹⁶³ Congress recognized the impracticality of such an arrangement, even with the relatively low numbers of patents in effect in 1842, and passed a law requiring inventors to mark their products with patents numbers to help the public know which patents covered which products. Patent holders thereafter faced criminal penalties for failing to mark.¹⁶⁴

The marking requirement was subsequently changed to limit damages in infringement cases rather than imposing a criminal penalty.¹⁶⁵ Patent holders have a duty to write “patented” and the patent number on all patented products they produce.¹⁶⁶ If a patent holder fails to mark his products in a “substantially consistent and continuous manner,” he is barred from collecting damages for any infringement before that point. The only exception is that if the infringer is given actual notice about the patent, at which point he can be sued for damages, regardless of whether the patent holder properly marked his products.¹⁶⁷

The marking must be on the article itself, or “when, from the character of the article, this can not be done, by fixing to it, or to the package wherein one or more of them is contained, a label containing a like notice.”¹⁶⁸ The word “article” in the statute has been interpreted to also include machines and other products.¹⁶⁹

If a patent holder licenses his patent to another manufacturer, that manufacturer must mark his products as well. If the licensee fails to do so, the patent holder is unable to collect damages from anybody, even if he marks his own products.¹⁷⁰ The duty is on the patentee to police his licensees.¹⁷¹ The reason for the marking duty is to protect the public from thinking that unmarked articles can be copied.¹⁷²

¹⁶³ “Patents are public records. All persons are bound to take notice of their contents. . . .” *Boyden v. Burke*, 55 U.S. 575, 582 (1852). The folly of relying on one central patent library was shown another way in 1836 when the building burned down.

¹⁶⁴ Act of Aug. 29, 1842, ch. 263, § 6, 5 Stat. 543, 544-45 (repealed 1861).

¹⁶⁵ Act of Mar. 2, 1861, ch. 88, § 18, 12 Stat. 249.

¹⁶⁶ 35 U.S.C. § 287(a) (2005). Patentees have the option of abbreviating “patented” as “pat.”

¹⁶⁷ *Nike, Inc. v. Wal-Mart Stores, Inc.*, 138 F.3d 1437, 1444, 1446 (Fed. Cir. 1998). In particular, it does not matter if the infringer merely knew about the patent, and it does not matter if notice was given by a third party. Notice given by the patent holder is required.

¹⁶⁸ 35 U.S.C. § 287(a) (2005).

¹⁶⁹ For example, the patent in *American Medical Systems, Inc. v. Medical Engineering Corp.* was titled as an apparatus and method, where “apparatus” normally means “machine,” but the “apparatus” claimed was merely a saline implant. 6 F.3d 1523, 1527.

¹⁷⁰ *SRI Int’l, Inc. v. Advanced Tech. Labs., Inc.*, 127 F.3d 1462, 1469 (Fed. Cir. 1997).

¹⁷¹ *Maxwell v. J. Baker, Inc.*, 86 F.3d 1098, 1111-12 (Fed. Cir. 1996). When licensees subcontract to third parties, the courts are more flexible with the marking duty, as long as the patent holder made reasonable efforts to ensure compliance. *Id.*

¹⁷² *Nike*, 138 F.3d at 1443.

A patent holder who sits on his patent, neither manufacturing himself nor licensing to third parties, is not required to mark anything or give actual notice before suing infringers.¹⁷³

2. Marking not required for process patents

Although patent holders have a duty to mark patented products sold on the open market, there is no such duty to mark products made by patented processes. More precisely, marking is not required when a patent is directed to a process.¹⁷⁴ For example, a new *kind* of high-strength steel could be a patented product, but ordinary steel made more cheaply by a new process could not. (It is the same steel, and patents cannot be granted on old products.) However, the *process* of making ordinary steel more cheaply could be patented; it is a new process.

The distinction is supported by the public policy behind the marking doctrine. Someone encountering a patented product may want to replicate it, and a marking on the product itself serves as a warning that they should not do so. To continue the steel example, a competitor may notice the strength of a new kind of steel. But how could the manufacturer making cheaper, ordinary steel mark the “process” when the resulting steel looks exactly the same?¹⁷⁵

When a patent is not directed to a process but is, nevertheless, cleverly drafted to contain process claims, the Federal Circuit has required marking. In *American Medical Systems*, the patented product was packaging for a pre-filled saline implant, and the patent also covered the process of putting the saline implant into the packaging.¹⁷⁶ The appeals court held that inventors could not evade the marking duty by drafting parallel process claims:

The purpose behind the marking statute is to encourage the patentee to give notice to the public of the patent. The reason that the marking statute does not apply to method claims is that, ordinarily, where the patent claims are directed to only a method or process there is nothing to mark. Where the patent contains both apparatus and method claims, however, to the extent that there is a

¹⁷³ *Wine Ry. Appliance Co. v. Enterprise Ry. Equipment Co.*, 297 U.S. 387, 397 (1936) (“We find nothing adequate to support the notion that such patentees were deprived of the right theretofore existing to claim damages from an infringer unless and until he could be run down and served with actual notice.”).

¹⁷⁴ *Am. Medical Sys.*, 6 F.3d at 1538 (“The law is clear that the notice provisions of section 287 do not apply where the patent is directed to a process or method.”).

¹⁷⁵ See *infra* note 173 (“nothing to mark” where processes are concerned). This public policy explanation unravels if tugged too hard. Some manufacturing processes can be reverse-engineered by examining the end products.

¹⁷⁶ The facts of the case strongly support applying the marking duty as a matter of public policy. The infringer obtained a sample of the product at a trade show. Because the sample was unmarked, the infringer believed it was unpatented and thus abandoned his own research efforts in favor of copying the superior packaging of the sample. See *supra* note 173, at 1528.

tangible item to mark by which notice of the asserted method claims can be given, a party is obliged to do so. . . .¹⁷⁷

A panel back in 1983 may have undercut *American Medical Systems*,¹⁷⁸ when it held that if only process claims were infringed, then marking is not required, regardless of how the original patent claims were drafted.¹⁷⁹ In *Hanson v. Alpine Valley Ski Area*, a patent on a snow machine included claims on the machine and the process of making snow. The machine was better than older snow machines and it was not marked as patented, so a competitor began making similar machines based on the same technology. The court held with little discussion that a process claim had been infringed, the apparatus claim was not at issue; therefore, damages were allowed.¹⁸⁰

B. Computer software and marking

To examine the relation of the marking doctrine to computer software, first it must be shown that computer software is a product. A computer program *per se* may not be patented, although it may be *copyrighted*.¹⁸¹ That is, it is not possible to patent Microsoft Windows just by mailing a CD-ROM to the patent office. Instead, software has to be claimed as a process, an article of manufacture, or a machine.¹⁸²

In more everyday terms, a “process” could be the steps a computer takes when running a program.¹⁸³ So for example, Microsoft could patent the ten steps in a new encryption algorithm. An “article of manufacture” is something like a floppy disk, containing a computer program. A “machine” (or “system”) patent on a computer program might be a computer configured to run it. These are all drafted in similar ways. When writing a patent application, it is easy to draft parallel claims by cutting and pasting, changing a few words here and there.¹⁸⁴

There is a reason that claims are written in different ways. Article claims are useful for suing competing manufacturers of computer software. As they produce CD-ROMs with the program, they directly infringe the patent, in the same way that

¹⁷⁷ *Id.* at 1538–39.

¹⁷⁸ At least one commentator thinks that the marking duty of *American Medical Systems* could be avoided by only asserting method claims during litigation. John R. Thomas, *Of Text, Technique, and the Tangible: Drafting Patent Claims Around Patent Rules*, 17 J. MARSHALL J. COMPUTER & INFO. L. 219, 256–57 (1998). This seems dubious.

¹⁷⁹ Joel Voelzke, *Patent Marking Under 35 U.S.C. § 287(a): Products, Processes, and the Deception of the Public*, 5 FED. CIR. B.J. 317, 332 (1995).

¹⁸⁰ 718 F.2d 1075, 1083 (Fed. Cir. 1983).

¹⁸¹ Dennis S. Karjala, *The Relative Roles of Patent and Copyright in the Protection of Computer Programs*, 17 J. MARSHALL J. COMPUTER & INFO. L. 41, 67–68 (1998).

¹⁸² Cohen & Lemley, *supra* note 160, at 20.

¹⁸³ See, for example, U.S. Patent No. 6,791,573 (issued Sept. 14, 2004), which has parallel article and process claims.

¹⁸⁴ The limiting factor may simply be the filing fees the client is willing to pay. Filing fees are based in part on the number of claims in a patent application.

any other manufacturer can infringe by producing a patented product.¹⁸⁵ Users can be sued under process claims because they are actually carrying out the steps described in the patent when they run the software.

Machine claims are not always included in software patents. Some patent attorneys see them as a relic from a time when the patent office would not allow article of manufacture claims.¹⁸⁶ Others write them hoping to increase damages in litigation. The theory as it is sometimes explained is that a system claim may be drafted to cover the entire machine on which the software is installed. For example, if an avionics manufacturer infringes a patent on just the software, the jury might award small royalties based simply on the software. But the royalties might be based on the whole airplane if that is what the patent claims.¹⁸⁷

If computer software is patented as an article or machine, then the marking doctrine applies, but if the claim is drafted as a process, then it may not need to be marked.¹⁸⁸ As a general rule, process claims do not require marking the invention, but it is doubtful that this loophole would apply to software patents in light of a recent Federal Circuit decision.

The Federal Circuit has already treated computer software as a product, in another area of patent law where process claims are treated more leniently than product claims. There is a general rule that a sale of a patented product more than a year before filing a patent will invalidate the patent—the so-called “on-sale bar.”¹⁸⁹ Licensing of processes is allowed without triggering the bar.¹⁹⁰ In *Minton v. N.A.S.D.*, the inventor argued that his software had not been on sale, rather he had “licensed” the patented method to a customer. The Federal Circuit held that computer software would be treated like any other tangible invention, even though the patent had been written with process claims.¹⁹¹ The appeals court was unwilling to allow the inventor to evade the on-sale bar by meaningless claim drafting formalisms. The on-sale bar,

¹⁸⁵ Article claims are relatively recent. During the 1990s, the patent office reversed itself and decided to accept computer program product claims. *In re Beauregard*, 53 F.3d 1583 (Fed. Cir. 1995).

¹⁸⁶ Gleick, *supra* note 121.

¹⁸⁷ True believers in the power of patents no doubt believe that these bigger infringement verdicts lead to more innovation. Empirical evidence suggests they are wrong. See James Bessen & Eric Maskin, *Sequential Innovation, Patents, and Imitation* (2000) (Department of Economics, Massachusetts Institute of Technology, Working Paper No. 00-01, 2000) (concluding patents restrict complementary innovation typical of fields like software, semiconductors, and computers); James Bessen & Robert M. Hunt, *An Empirical Look at Software Patents* (Research on Innovation, Working Paper No. 03-17/R, 2004) (concluding growth of software patents is largely strategic, corresponding to *decrease* in R&D).

¹⁸⁸ See *supra* text accompanying notes 173–79.

¹⁸⁹ *Pfaff v. Wells Electronics, Inc.*, 525 U.S. 55, 57 (1998). The on-sale bar is peculiar to the patent law of the United States.

¹⁹⁰ *In re Kollar*, 286 F.3d 1326, 1332 (Fed. Cir. 2002).

¹⁹¹ *Minton v. Nat'l Ass'n of Securities Dealers*, 336 F.3d 1373, 1378 (Fed. Cir. 2003).

like the marking duty, is put in place to prevent the public from thinking that patented products are free to use.¹⁹²

It is unclear how software must be marked. The literal reading of § 287(a) states that the “article” must be marked unless for physical reasons it is impossible to do so.¹⁹³ For a claim on a computer-readable medium (e.g. a floppy disk or CD-ROM), the disc is the medium, and a literal reading of the law and the claim would therefore seem to say that the disc itself must be marked with each patent used. Once software is installed, the software must continue to be marked, as a hard drive usually still falls within the letter of the claim. It is practically impossible, however, to ensure the marking of a hard drive.¹⁹⁴ The second best alternative is to display the patent number on start-up. Some companies, such as Adobe, already make efforts to comply with the marking duty by marking the physical media and displaying patent numbers as the programs start.¹⁹⁵ There does not appear to be any Federal Circuit precedent on how software must be marked. The Silicon Valley law firm Fenwick & West LLP suggests this practice of marking media and splash screens.¹⁹⁶ Some companies, like Nintendo of America and NetObjects, list patents in the user manual, although this seems less likely to satisfy the marking duty. The Adobe approach seems fair to both the patent-holder and the end-user, and complies with the letter and spirit of the present law.

If there are so many patent numbers applying to the computer program that they cannot be printed legibly on the CD-ROM, then presumably they could be listed on the paper insert of the jewel case. It seems unlikely, however, that a company would mark its software with so many patent numbers. If it erroneously marked its software with inapplicable patent numbers, it would face steep civil penalties.¹⁹⁷

C. Towards a stronger, more uniform marking duty

¹⁹² *In re Mahurkar Double Lumen Hemodialysis Catheter*, 71 F.3d 1573, 1577 (Fed. Cir. 1995).

¹⁹³ See *supra* text accompanying note 167.

¹⁹⁴ Literally marking the computer-readable medium would mean instructing the user to disassemble his computer and put a sticker on his hard drive. Clearly this would not be practical.

¹⁹⁵ The start-up, “splash” screen of Adobe Elements 2.0 lists a number of patents. The patents are also listed on the compact disc installation medium. Other Adobe software does the same thing. The company may be trying to protect its most valuable patents.

¹⁹⁶ Fenwick & West LLP, Patent Protection: For High Technology and Life Sciences Companies 11 (2002), *at* http://www.fenwick.com/docstore/publications/IP/Patent_Protection.pdf.

¹⁹⁷ There is a fine of \$500 for each false marking “offense,” although it is unclear what constitutes an “offense.” Any private citizen can sue for the penalty, in which case half of the money goes to the United States. 35 U.S.C. § 292 (2005).

The marking duty almost certainly applies to software patents today, but it is rarely fulfilled. A stronger marking duty would bring patentees of software in line with the practices of other kinds of patent-holders.

1. The present duty to mark software: Rarely fulfilled

If the software is distributed in a tangible form, the inventor has a duty to mark the physical medium; in normal situations it is insufficient to note the patent number on the packaging or to have the program display the number while running. That is, the Adobe approach is probably correct, although a software company that merely listed its patents in the user manual would probably put up a fight if its practice were challenged in court. The Adobe practice follows from the letter of the law: the “article” must be marked unless that would not be practicable.

Software must be marked regardless if the claims are written as processes, articles, or machines. There is precedent in the Federal Circuit for grouping process claims with apparatus claims for purposes of the marking duty,¹⁹⁸ and there is precedent for treating software process claims as product claims to prevent inventors from escaping statutory requirements governing tangible inventions.¹⁹⁹

How the marking duty should be enforced where no physical medium exists has not yet been discussed either in case law or in law review articles. At least with software distributed electronically, a physical medium exists once the user installs it. Online services are more difficult to contemplate; the patented computer program runs on a server locked up far away. To use PriceLine’s ticket auction as an example, even though the basic idea is readily apparent to anyone using the service, there does not appear to be any requirement that the public be put on notice as to which patents cover those algorithms.²⁰⁰ One could imagine almost any kind of software being run as an online service, tantalizing potential infringers, outside the ambit of the marking duty.

It appears that software is rarely marked. There are a handful of published orders where the parties agreed that marking did not occur.²⁰¹ The lack of other published cases on point suggests that parties in software patent litigation tend to stipulate early in the proceedings that the marking duty had not been met. This

¹⁹⁸ See *supra* text accompanying notes 175–179.

¹⁹⁹ See *supra* text accompanying notes 188–191.

²⁰⁰ In practice, however, PriceLine has only one main product, and so it is reasonable to simply read all of the patents assigned to the company in chronological order. If the ticket auction had instead been provided by Microsoft through its MSN online service, it would be much more difficult to figure out the degree of patent coverage. It would be particularly nice to have a marking requirement for so-called “business method” patents, but it is unclear how one would formulate it.

²⁰¹ In these cases § 287(a) was satisfied only by actual notice. *AT&T Corp. v. Microsoft Corp.*, 290 F. Supp. 2d 409, 412 (S.D.N.Y. 2003); *Articulate Systems, Inc. v. Apple Computer, Inc.*, 53 F. Supp. 2d 78, 79 (D. Mass. 1999).

habitual stipulation in turn suggests that, generally speaking, no attempt is made to mark computer software.

The effective rule for damages in software infringement cases, therefore, is that damages are not recoverable before actual notice unless the inventor has chosen not to capitalize on his invention, in which case the inventor can collect damages for the full period of infringement. The perverse consequence is that an inventor of an implementation that failed in the market will be able to get more money in litigation than one who continues to make a good product.²⁰²

2. A stronger marking requirement

Long ago inventors would be criminally punished for failing to mark their products, but then the law was changed to couple the punishment (no retrospective damages) with patent law. It seems that a more severe punishment is necessary to ensure that software products continue to be marked. A stronger marking duty would provide that patent holders be permanently barred from enforcing patents on products not marked. Concerns about deciding which patents are “software patents” could be met by uniformly applying the heightened duty to all patents on articles.²⁰³ The Adobe standard would suffice for most kinds of software. It is unclear what to do about online services.

A milder proposal is that damages would be disallowed until the products are marked, regardless of notice. In practical terms it might amount to the same thing. Major computer patent portfolios are so large that companies don’t know what they own,²⁰⁴ and it seems unlikely that any large-scale marking effort would take place given the expense and the need for near-total compliance among the companies’ own products *as well as those of licensees*. It seems more likely that marking efforts would focus on the most valuable patents, which have already been identified for licensing purposes.²⁰⁵

²⁰² An example of such a failed invention is the patent that was asserted against Lotus. See *supra* note 127 and accompanying text.

²⁰³ In *American Medical Systems*, a research and development project was canceled because a competitor’s design was believed to be in the public domain. The infringer clearly suffered detrimental reliance. If the rule were a permanent bar on damages, then the innocent infringer would not have had to pay massive litigation costs and pay to redevelop what it had already canceled. See *supra* note 168, at 1528.

²⁰⁴ Posting from Bruce Perens, formerly at bruce@pixar.com, to debian-legal@lists.debian.org (Dec. 7, 1998) (companies like I.B.M. have patented so much that they don’t know what their patents cover, citing discussions with I.B.M. legal regarding I.B.M.’s first open-source software license) (on file with STAN. J.L. BUS. & FIN.).

²⁰⁵ See *infra* note 214.

Congress would be best situated to make this proposed change. The Federal Circuit could also make it equitably as a form of estoppel consistent with the spirit of the marking duty, given its near-total control over the interpretation of patent law.²⁰⁶

D. Collateral impacts of marking software

1. Disclosure

The main effect is that anyone could determine which patents cover what software products by looking at the list of patents marked. Therefore, the economic incentive benefits of patents requiring widespread disclosure would ensue: the incentives to invent, design around, disclose, and invent.²⁰⁷

There are three acknowledged limitations to the benefits. First, someone must examine a product to learn of the patents by marking. For example, Quantel could still have sued Adobe if it had properly marked its video equipment.²⁰⁸ Adobe might never think to check a product in a different line of business for patents it might infringe.

Next, the marking requirement would still not apply to patent owners who do not make products. If Quantel had never produced anything, existing only as a patent litigation shell, Adobe would have been even less likely to uncover the patents. Perhaps a solution to these first two problems lies in controlling the scope of software patent claims and requiring better technical disclosures. Arguably such reforms are sorely needed.²⁰⁹

Lastly, tightening the rules for marking patents may amount to a “taking” requiring compensation to the patent owners. Such a finding seems unlikely in view of the requirement that the patent owner first prove a denial of “all economically beneficial or productive use.”²¹⁰ By complying with the marking requirement (which has been on the books as a duty in one form or another for a hundred and fifty years), patent owners would continue to benefit as they did before. Presumably they would be given a grace period of year or two after the law changed, too, for coming into compliance.

2. Start-ups

The other benefit of patents listed in the first part of this paper, solving Arrow’s Information Paradox, would be preserved. This proposal would not hurt

²⁰⁶ As the Federal Circuit was wholly responsible for allowing software patents in the first place, *see supra* note 151–60, it is at least arguably appropriate for it to take action to fix the problem it created.

²⁰⁷ *See supra* text accompanying notes 16–70.

²⁰⁸ *See supra* text accompanying notes 89–90.

²⁰⁹ *See supra* note 47.

²¹⁰ *Lucas v. S.C. Coastal Council*, 112 U.S. 2886, 2897–98 (1992).

start-ups who use patents in their negotiations with larger companies. As long as a company diligently marks its software, it doesn't need to worry that it will lose its patent through the marking requirement. Because it would only have a handful of products or patents, it is relatively easy to determine the scope of its patents and mark the products accordingly.

The law as it stands has the advantage that if a small company accidentally neglects to mark some of its products (for example, early releases distributed at a trade show), the patents aren't lost forever. On the other hand, there is the disadvantage that recipients of the software may assume that they can implement some of the ideas, not knowing that they are covered by patents. The exact situation of prototypes being shown at a trade show, where the finished product was later marked, led to litigation in *American Medical Systems*.²¹¹ Plus, if an unmarked release had truly been inconsequential, the law would forgive the transgression.²¹²

3. Blanket licensing

One of the most peculiar innovations in patent practice over the last few decades has been the emergence of blanket patent licensing of gigantic patent portfolios as an alternative to licensing individual patents. The practice is often described as "cross" licensing. The term "cross-licensing" is technically true but misleading.²¹³ When a very large company cross-licenses with a start-up, although both companies license their portfolios to each other, the larger company usually has quite a bit more on the table.²¹⁴ It may demand cash payments, equity, or other forms of compensation to make the deal even. Reportedly I.B.M. categorizes its patents into "groups." Group 1 patents cost 1% of a licensee's gross sales. Group 2 patents cost

²¹¹ See *supra* note 168.

²¹² *Maxwell v. K Mart Corp.*, 880 F. Supp. 1323, 1336 (D. Minn. 1995).

²¹³ Some authors have stated that the word "cross-licensing" carries implications as to whether money changes hands as royalties or fixed payments. Mark A. Lemley, *Rational Ignorance at the Patent Office*, 95 NW. U. L. REV. 1495, 1505 (2001).

²¹⁴ *Id.* at 1504. The point was made in a more concrete way by Hall and Ziedonis:

Some noted that the semiconductor industry has historically been characterized by broad cross-licenses of patent rights among manufacturers. To a large extent, this is still the case. Nonetheless, a firm lacking a strong patent portfolio of its own with which to negotiate licensing or cross-licensing agreements could face a more rapid erosion of profits. . . . For example, one industry executive estimates that "a new manufacturer would need to spend \$100-\$200 million of revenues to license what are now considered basic manufacturing principles but which do not transfer any currently useful technologies."

Hall and Ziedonis, *supra* note 134 (citations omitted). Many of the economic papers cited in this article point out that semiconductor patents and software patents are similar from an economic standpoint.

2% each. The entire I.B.M. portfolio can be licensed for 5% of a licensee's gross sales.²¹⁵

In these cross-licensing deals, neither party is exactly sure what they are licensing or what products infringe.²¹⁶ Instead the entire portfolios are cross-licensed, sometimes excluding a short list of exceptionally valuable patents. Not surprisingly, when patent portfolios of thousands of patents are licensed en masse, the marking duty is not universally enforced. How could licensees, after all, know the scope of a patent portfolio when the owner himself does not?²¹⁷ In at least one I.B.M. contract the marking duty is not mentioned at all.

Marking is incompatible with blanket licensing of large numbers of product patents.²¹⁸ Checking to see whether a patent covers a particular product is very expensive, potentially costing tens of thousands of dollars per patent per product.²¹⁹ Even if the cost were driven down through organizational efficiency, it would cost tens of millions of dollars to evaluate a large product line against a large patent portfolio. A company may be willing to invest money checking its own products to preserve its patents. Yet if it blanket licensed its portfolio, the company would need to check all of the *licensee's* products as well, and if the licensee fails to mark, the company is hit with the penalty of essentially forfeiting the unmarked patents. Moreover, given that a license fee may "only" be a few million dollars, it may not even be profitable to diligently check the licensee's products for infringement against the entire portfolio.

It seems that the natural result would be that the bulk of trivial patents would be abandoned as being unprofitable to license, and the most valuable patents would be licensed individually. It appears that large companies already sort their patent portfolios by worth, like how I.B.M. is reported to have Group 1, Group 2, and strategically licensed patents.²²⁰ There would be less incentive to harvest patents from ordinary research and development, as well.

Why are blanket licenses controversial? A blanket patent right is not a technology transfer. If a competitor wants to implement or even just learn from a patented idea, a mere cross-license won't come with any information not already

²¹⁵ Paul Heckel, *Debunking the Software Patent Myth*, COMMUNICATIONS OF THE ACM, June 1992, at 134. The information may be slightly out-of-date, although it is generally consistent with second-hand reports and other sources. Currently it appears that a few particularly valuable patents are excluded from I.B.M. blanket licenses.

²¹⁶ *Id.* at n.42.

²¹⁷ See *supra* note 203.

²¹⁸ Blanket licensing would still occur with genuine process patents, as in the semiconductor industry, because marking is generally not required for process claims.

²¹⁹ See *supra* text accompanying notes 131–35.

²²⁰ See *supra* note 214 and accompanying text.

available in the public patents,²²¹ or even tell the competitor which patents cover the idea.

Sometimes small companies enter into blanket licensing arrangements to obtain rights to particular advanced technology.²²² Other times they simply want to remove the cloud of a ruinous lawsuit. In a sense, both are “benefits” accruing to the smaller firm from blanket licensing. On a macroscopic level, however, does extensive cross-licensing really benefit society?²²³ If one accepts the propositions that small start-up companies are the engines of innovation and that the patent system is intended to further innovation, it seems curious that patents are used as a kind of tax, funneling money from start-ups into the biggest computer companies. If the “tax” is on revenues, not profits, entering into a cross-licensing arrangement with just one large company could bankrupt a start-up.

As discussed previously,²²⁴ blanket licensing and patent harvesting, or the filing of patents in the course of ordinary product development, in principle increases the financial returns to research. But economic analysis has shown that software patents, harvested or not, *do not* lead to an increased willingness to undertake research with lower expected returns.²²⁵

Given the billions of dollars changing hands each year on account of blanket licenses and the additional indirect deadweight costs, the onus is on the proponents of patent harvesting to show where the offsetting benefits to society can be found.²²⁶

²²¹ Normally blanket licenses do not cover any transfer of technology or know-how. I.B.M. Worldwide Patent Licensing Practices, at <http://www.ibm.com/ibm/licensing/patents/practices.shtml>. See also *supra* note 213.

²²² I.B.M.’s Q-coder, for instance, has been widely described in the literature but is covered by patents, for example: U.S. Patent Nos. 4,286,256 (issued Aug. 21, 1981), 4,295,125 (issued Oct. 13, 1981), 4,463,342 (issued July 31, 1984), 4,467,317 (issued Aug. 21, 1984), 4,633,490 (issued Dec. 30, 1986), 4,652,856 (issued Mar. 24, 1987), 4,891,643 (issued Jan. 2, 1990), 4,905,297 (issued Feb. 27, 1990), and 4,935,882 (issued June 19, 1990). A firm might well choose to license I.B.M.’s entire portfolio because it is cheaper than licensing the patents individually. See *supra* note 214.

²²³ One could argue that the benefit of a blanket license is often as illusory as the benefit of paying “protection” money to the mafia, as both serve the same purpose. It seems a little hypocritical when infringement of these unknowably obscure software patents is condemned as “plagiarism” or “piracy.” See, e.g., Patent Protection for Software-Related Inventions: Hearing Before the Patent and Trademark Office, Dept. of Commerce (1994) (statement of Richard LeFaivre, Apple Computer).

²²⁴ See *supra* notes 24–25 and accompanying text.

²²⁵ See Bessen & Hunt, *supra* note 186, at 16–20.

²²⁶ Such analysis would be fairly easy given the availability of recent comparative economic data. Patent harvesting and vigorous cross-licensing at I.B.M. began with Marshall Phelps’ tenure at I.B.M. in the 1980s. Broad licensing in the semiconductor industry began even later with Texas Instruments. See *supra* note 134.

Blanket licensing arose out of a desire to increase revenue, not as a way of minimizing litigation. In fact patent litigation has dramatically *increased* over the past few decades.

4. Patent poaching

Ideally a stronger marking requirement would put an end to litigation by patent owners who exit the software market, wait many years, and then sue the leading innovators. (For the sake of brevity, this practice will be referred to as “patent poaching.”²²⁷) Although this proposal would mitigate patent poaching, it would not be able to end it completely.

There are two ways that patent poaching would be mitigated. The first is that a company would be barred from retrospectively deciding that a patent is worth enforcing after competitors come to rely on the invention being in the public domain.²²⁸ The most prominent example in recent years of patent poaching by a large computer company is Unisys and its patent on Lempel-Ziv-Welch data compression, better known as the “GIF patent.” The inventors published the algorithm in *IEEE Computer* magazine in June 1984.²²⁹ Unbeknownst to the public, they had also filed for a patent. In the meantime the algorithm became incorporated into a few popular file formats, including the graphics format GIF (1987) and the UNIX general-purpose format *compress* (1986).²³⁰ The former is one of the two main graphics formats in use on the World Wide Web.

Unisys waited to assert the patent until 1994, when the Internet boom was beginning, and made progressively larger efforts to collect royalties from GIF users in 1995 and 1999.²³¹ If this article’s proposed marking requirement had been in effect, Unisys would have estopped itself from enforcing the GIF patent when it shipped UNIX computer systems in the mid-1980s. If Unisys had valued the patent enough to mark the software, the public would have immediately been put on notice that a different algorithm should be used to avoid patent liability.²³²

²²⁷ The term “submarine patents” is sometimes used, but it is better reserved for patents which age for years in the patent office before issuing. *See, e.g.,* Symbol Tech. v. Lemelson Medical, 277 F.3d 1361, 1370 n.1 (Fed. Cir. 2002) (Newman, J., dissenting). A similar phenomenon, where patented technologies are secretly injected into industry standards, has been called “patent farming.” Bruce Perens, *The Problem of Software Patents in Standards*, at <http://perens.com/Articles/PatentFarming.html> (Nov. 8, 2004).

²²⁸ As mentioned *supra* notes 38–42 and accompanying text, software has network effects, and there are significant social costs to asserting patents after a network has grown.

²²⁹ Terry A. Welch, *A Technique for High Performance Data Compression*, *IEEE COMPUTER* 8–19, June 1984.

²³⁰ *Compress* appeared in the 4.3BSD version of UNIX, which was released in June 1986. Compress(1) Manual Page, at <http://developer.apple.com/documentation/Darwin/Reference/ManPages/man1/compress.1.html> (last updated 1994) (first appearance of *compress*); PETER SALUS, *A QUARTER CENTURY OF UNIX 165* (1994) (release of 4.3BSD).

²³¹ Free Software Foundation, *Why There Are No GIF files on GNU Web Pages*, at <http://www.gnu.org/philosophy/gif.html> (last updated June 21, 2004).

²³² Perhaps the algorithm could have been changed while the network was still small. *See supra* note 227.

The other way a strengthened marking duty would mitigate the poaching problem would be by stopping companies from taking expansive views of their patent portfolios years after the patents are granted. The most prominent example coincidentally relates to the other graphics file format in widespread use on the Internet as well as in digital cameras—JPEG. The format, formalized in 1990,²³³ was intended to be in the public domain in its most basic implementation. In 2002 a small company named Forgent asserted that one of its patents, issued in the 1980s, covered part of the process of making JPEG files.²³⁴

Forgent would have been prevented from unsettling the status of the JPEG standard. It is likely that Forgent, being in the videoconferencing business, shipped software using JPEG files sometime between 1990 and 2002. It could not have marked the software before believing that the patent covered the JPEG standard.²³⁵ When it asserted the patent claim against JPEG, it would have rendered the patent permanently unenforceable. In this way the proposed marking requirement sets out for the public not only what is patented, but also what is not patented.²³⁶

5. Free software

One might think of software as being like soap: a product for sale. In fact, because there is no cost to duplicating software, there is an increasing amount of software that has been developed with the intent of giving it to the world at no cost, without the author retaining control over the design documents, or “source code.” Software patents are potentially a real threat to free software, but with this paper’s proposal the threat would be mitigated.

Perhaps the most famous free software product is the operating system GNU/Linux,²³⁷ the kernel of which was originally created by a Finnish university student and is now jointly developed by people ranging from volunteer individuals to the biggest computer companies in the world.²³⁸ Bill Gates has said that several years hence, GNU/Linux will be the only remaining competitor to Microsoft Windows.²³⁹ Although the market share of servers with GNU/Linux is unclear, it is

²³³ World Wide Web Consortium, *JPEG*, at <http://www.w3.org/Graphics/JPEG/> (last updated Jan. 9, 2003).

²³⁴ See *supra* text accompanying notes 136–138. The Forgent patent only covered a tiny piece of the JPEG standard.

²³⁵ There are steep penalties for false marking (enforceable privately or by the government), see *supra* note 196, so a company would not mark a product unless it truly believed that the patent covered the product.

²³⁶ See *supra* text accompanying note 70.

²³⁷ Other examples of operating systems are Microsoft Windows running on PCs, Apple OS X running on Macintoshes, and UNIX running on engineers’ workstations.

²³⁸ The remainder of the operating system is GNU software, under the auspices of the Free Software Foundation. Several important parts of GNU/Linux predate the Linux kernel.

²³⁹ Paul Krill, *Gates Undaunted by Linux*, INFOWORLD, Oct. 1, 2004, at http://www.infoworld.com/article/04/10/01/HNgatesstalksmuseum_1.html (last visited Apr. 25, 2005).

certainly very large.²⁴⁰ Clearly customers and computer companies put significant value on GNU/Linux.²⁴¹

Yet software patents could discourage users from taking advantage of the software. One study found 283 patents that the Linux kernel likely infringes.²⁴² A leaked Hewlett-Packard executive memo from 2002 states: "Microsoft intends to sue companies shipping Free and open source programs that potentially violate their patents."²⁴³

For GNU/Linux and other free software, software patents are a uniquely serious problem. Any product sold for money (e.g., soap) has revenue, which can be used to pay for patent licenses. Likewise revenue can be spent on patent applications in order to obtain the economic benefits of patents listed in the first part of the paper. Free software, on the other hand, often has no revenue for licenses and does not benefit from patents. When royalties are required for patents, free software is unable to implement industry standards despite the betterment of society by having free implementations. For this reason the World Wide Web Consortium demands that its standards be available without royalties.²⁴⁴

If the proposal in this paper were implemented, much of the problem with patents and free software would disappear. Many major computer companies have redistributed free software projects. For example, server companies sell computers with GNU/Linux pre-installed. By continuing to do so under the proposed marking requirement, they would permanently be estopped from asserting any of the patents

²⁴⁰ *Id.* Bill Gates contested the notion that GNU/Linux servers are a majority of the market, but if they are not a majority then they are certainly a significant percentage. Most major server software, with the principal exception of Microsoft's server products, is now available for GNU/Linux.

²⁴¹ Reasons often cited for choosing GNU/Linux include lower maintenance costs, better performance, support for more types of hardware, conformance to industry standards, stability, security, better leverage of I.T. workers, lower up-front software costs, etc.

²⁴² Dan Ravicher, *Mitigating Linux Patent Risk* (Aug. 2, 2004), at http://www.osriskmanagement.com/pdf_articles/linuxpatentpaper.pdf. Although Ravicher may seem biased, as a purveyor of insurance to cover GNU/Linux patent risk, his study was seen as legitimate by prominent free software thinkers. E-mail from Richard Stallman, founder of the Free Software Foundation, to the author (Sept. 13, 2004) (on file with the STAN. J.L. BUS. & FIN.). The chairman of Microsoft has also publicly cited the study. Steve Ballmer, Speech to Asian Government Leaders Forum (Nov. 18, 2004). Although the official Microsoft transcript (at <http://www.microsoft.com/presspass/exec/steve/2004/11-18GLF.asp>) does not mention Linux, Reuters reported the remarks the same day under the headline, "Microsoft Warns Asian Governments of Linux Lawsuits." Reuters, *Microsoft Warns Asian Governments of Linux Lawsuits* (Nov. 19, 2004) (on file with the STAN. J.L. BUS. & FIN.).

²⁴³ Ravicher, *supra* note 241. The distinction between free and open-source software is not relevant for the purposes of his paper.

²⁴⁴ World Wide Web Consortium Patent Policy (Feb. 5, 2004), at <http://www.w3.org/Consortium/Patent-Policy-20040205/>.

infringed by that project,²⁴⁵ whether or not they originated the part of the software in question.²⁴⁶

IV. Conclusion

This article argues that software patents could at least be brought in line with other patents by strengthening the marking duty. Software patents are largely unjustifiable in the absence of marking.

There are five economic benefits to patents described in the literature: the incentive to conduct research on new projects, the incentive to design competing products, the incentive to disclose trade secrets so that others may benefit, the solution to Kenneth Arrow's information paradox, and the incentive to invest the money necessary to bring inventions to market. All of these depend in one form or another on public knowledge of parts of the relevant patents—either the “claims,” which state what the statutory monopoly actually covers, or the “specification,” the mandatory technical disclosure of trade secrets.

Unfortunately, it is hard for the public to figure out which software patents cover which products. In areas like chemical engineering, it is relatively easy to search in the patent libraries to find relevant patents. In mechanical engineering, the patent numbers are marked on the products themselves. Software patents do neither. They are nigh impossible to find in the patent libraries, and few companies mark their software with their patent numbers.

Perhaps the easiest solution is to put more pressure on companies to mark their software patent numbers. Aside from the direct benefits listed above, there would also be economic gains to be found from reducing blanket licensing and patent harvesting, preventing unsavory companies from coming out of the woodwork to assert patents against long-established industry standards, and making the legal landscape more conducive to free software like GNU/Linux.

One might object to the arguments presented in the second section of this paper as empirically unsupported—after all, the plural of “anecdote” is not “data.” The rejoinder is that a half-billion dollar verdict is more than a mere anecdote,²⁴⁷ and the plural of these “anecdotes” is a shameful abomination. The burden is on the

²⁴⁵ Presumably these patents would be the ones of most concern to free software developers. Ravicher's study, *supra* note 241, found that most patents infringed by the Linux kernel were held by major computer companies.

²⁴⁶ Although the computer company might take pains to mark the software that it distributes, it has no control over end users. Under free software licenses, users are licensed to modify and redistribute software. It is unlikely that all licensees, sub-licensees, sub-sub-licensees, etc. would continue to mark the software. Patent owners have the obligation to ensure marking by licensees, and by failing this obligation they would be permanently estopped from asserting those patents by a strong marking requirement.

²⁴⁷ Microsoft has been ordered to pay, or has settled to pay, several verdicts of this size just in the past two years. See *supra* note 13.

proponents of the current software-patenting regime to point out where the billions of economic gains can be found.²⁴⁸

In the absence of such a finding, a strengthened marking duty would go a long way towards addressing the concerns about software patents.

²⁴⁸ See Lessig, *supra* note 20 (“economists have found it very hard to reckon any net benefits” of software patents, which have “produced a whole cottage industry of shysters”).