

SOFTWARE PATENTS: INNOVATION OR LITIGATION?

Linda Levine¹, Kurt M. Saunders²

¹ *Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213 (USA), ll@sei.cmu.edu*

² *College of Business and Economics, California State University, Northridge, CA 91130 (USA), kurt.saunders@csun.edu*

Abstract: The Directive on the Patentability of Computer-Implemented Inventions recently approved by the European Parliament may have significant implications for the software industry, public policy and patent protection. In this paper, we summarize the scope of patent protection in the European Union, the United States, and Japan. We examine the patentability of computer software under E.U. and U.S. patent law and also consider two instances of software patenting and their effects. We provide an overview of the Directive and finally assess the legal, economic, and public policy implications for software developers and users.

Key words: software patents, scope of patent protection, patentability of computer-implemented inventions

1. INTRODUCTION

The Directive on the Patentability of Computer-Implemented Inventions recently approved by the European Parliament may have long range implications for the software industry, public policy and patent protection for years to come. Patents can be, at once, a spur and a roadblock to innovation. At the same time, the world is moving toward greater economic integration, driving current debate on the harmonization of intellectual property rights.

In this paper, we begin by summarizing the scope of patent protection in the European Union, the United States, and Japan. In doing so, we examine the patentability of computer software as inventions allowed under E.U. and U.S. patent law. The value of software patents and the U.S. experience with patent liberalization for 20 years is considered. Two instances of software patenting and their effects are discussed. We then provide an overview of the proposed Directive recently approved by the European Parliament. The approval also included several amendments. Finally, we assess the legal, economic, and public policy implications of the Directive for software developers and users.

We do not attempt a comprehensive examination of the utility and rationale for patent protection. This is beyond the scope of our paper. Ours is also not an empirical research study; rather, our investigation is more contemporary and topical, looking at the conditions surrounding software patentability as a key barrier and enabler for innovation and competition.

2. PATENT LAW AND COMPUTER SOFTWARE IN THE EUROPEAN UNION

The European Patent Convention (EPC) serves as the basis for a harmonized system of patent protection for all members of the European Union. European patents have the same effect as patents granted by each nation under its own national patent laws. Article 52(1) of the EPC states: “European patents shall be granted for any inventions which are susceptible of industrial application, which are new and which involve an inventive step.” Article 52(2)(c) of the EPC specifically excludes “methods for ... doing business, and programs for computers” from the definition of inventions eligible for patent protection.

Although Article 52(2)(c) appears to completely exclude computer programs and business methods from patentability, Article 52(3) qualifies this exclusion by stating that the exclusion only applies to the extent an invention “relates to such subject matter or activities as such.” This suggests that while computer programs standing alone are unpatentable, the presence of a computer program as one component of a larger invention does not prevent the invention from qualifying for a patent. The European Patent Office (EPO) has likewise taken this position in its *Guidelines of the European Patent Office*, which explains: “A computer program claimed by itself or as a record on a carrier, is not patentable irrespective of its content. ... If however the subject-matter as claimed makes a technical contribution to the known art, patentability should not be denied merely on the ground that a computer program is involved.”

The leading case of the EPO Technical Board of Appeal involving software patentability is *Computer-related Invention/VICOM* (1986), which involved a program for digitally processing images. The Board held that the program was patentable since it was related to a technical process leading to a change in the physical images and was not merely a mathematical algorithm that manipulated numbers to calculate a purely numerical result. As to computer-implemented business methods, the key decision was *Pension Benefit Systems* (2001), where the application claimed a method for controlling a pension benefits program. The Technical Board of Appeal affirmed that the method was unpatentable, stating that “methods only involving economic concepts and practices of doing business are not [patentable] inventions.”

3. THE SCOPE OF PROTECTION FOR SOFTWARE IN THE UNITED STATES AND JAPAN

Under section 101 of the U.S. Patent Act, subject matter that may be patented includes any “process, machine, manufacture, or composition of matter, or any ... useful improvement thereof.” In addition, the invention

must be useful, novel, and nonobvious to qualify for a patent. To be novel and nonobvious, the invention must be new in that it is not part of the field of existing technology or “prior art,” and it must represent an inventive next step beyond the prior art, rather than an obvious variation.

Before the 1980s, it was generally accepted in the U.S. that software was not patentable. However, in 1981 the U.S. Supreme Court decided the case of *Diamond v. Diehr*, which involved a computer program, applied to a process for curing rubber, and expressly held for the first time that computer software was patentable so long as the claimed invention was not merely a procedure for solving a mathematical formula.

In 1998, the U.S. Patent & Trademark Office (USPTO) issued its *Examination Guidelines for Computer-Related Inventions* for its examiners to use in evaluating the patentability of software. Essentially, the USPTO Guidelines make clear that any type of software or computer-implemented invention is eligible for patent protection if the other tests for patentability – utility, novelty, and nonobviousness – are also met. The Japanese Patent Office (JPO) has adopted a similar position in its *Guidelines for Computer Software Related Inventions*. Under the JPO Guidelines, computer programs are patentable in Japan as long as they possess a high degree of technical creativity and utility.

Also in 1998, the U.S. Court of Appeals for the Federal Circuit, in the case of *State Street Bank & Trust Co. v. Signature Financial Group, Inc.* (1998), upheld as valid a patent directed to a computer-implemented business method designed to perform financial calculations and data-processing for mutual fund investments. This case was important because prior to this decision, it was widely believed that business methods and systems were not patentable. The *State Street* case made clear that business methods were to be evaluated in the same manner as any other type of process. In 2000, the JPO followed suit and revised its Guidelines to allow for the patenting of computer-implemented business methods when there is clear “involvement of inventive step.”

Since 1976, the USPTO has granted a steadily increasing number of patents for software-related inventions (see Appendix). Presently, 15% of all patents granted in the U.S. are software patents and the growth in software patents accounts for over 25% of the total growth in the number of patents issued between 1976 and 2001. Moreover, one recent study has reported that: “Overall, software patents are more likely to be obtained by larger firms, established firms, U.S. firms, and firms in manufacturing (and IBM); they are less likely to be obtained by individuals, small firms, newly public firms, foreign firms, and software publishers” (Bessen & Hunt, 2003, p.9).

Bessen and Hunt found that the correlation between R&D and patenting in the U.S. over time has been significantly *negative*. In other words, as software patenting rates have risen, R&D investment in sectors using information technologies has declined. This is not to say that other confounding factors may not have contributed to this phenomenon. We comment more extensively on the consequences of patent liberalization in the Reassessment section.

4. THE COMPLEX REALITY OF SOFTWARE PATENTS: TWO EXAMPLES

In this section, we examine two instances of software patenting and their effects. One of the patents led to litigation and one led to settlement. Both illustrate some of the unforeseen consequences that follow a decision to obtain a patent on software and then to enforce it.

4.1 “One Click” Shopping

The recent case of *Amazon.Com, Inc. v. Barnesandnoble.Com, Inc.*, No. 00-1109 (Fed. Cir. 2001) demonstrates the real-world implications of patenting software and business methods. The case involved U.S. Patent No. 5,960,411 (“the ‘411 patent”), which was issued to the inventor/programmer on September 28, 1999, and later assigned to Amazon. On October 21, 1999, Amazon sued BarnesandNoble.com (BN) alleging infringement of the patent.

Amazon’s patent was directed to a method and system for “single action” ordering of items in a client/server environment such as the Internet, known as the “One-Click” shopping model. The ‘411 patent described an approach in which a consumer could purchase items via an electronic network using only a “single action,” such as the click of a computer mouse button. Amazon developed the patent to deal with what it considered to be the frustrations presented by the “shopping cart model” for online purchasing. This method is described in the following excerpt from the ‘411 patent:

1. A method of placing an order for an item comprising:
under control of a client system, displaying information identifying the item; and
in response to only a single action being performed, sending a request to order the item along with an identifier of a purchaser of the item to a server system;
under control of a single-action ordering component of the server system, receiving the request;
retrieving additional information previously stored for the purchaser identified by the identifier in the received request; and
generating an order to purchase the requested item for the purchaser identified by the identifier in the received request using the retrieved additional information; and
fulfilling the generated order to complete purchase of the item whereby the item is ordered without using a shopping cart ordering model.
2. The method of claim 1 wherein the displaying of information includes displaying information indicating the single action.

Ultimately, the court ruled that Amazon demonstrated that it would likely succeed in its infringement suit. However, the court also observed that BN had raised substantial questions as to the validity of the ‘411 patent based on obviousness. As such, the court returned the case to the district court for further review of the patent and the alleged infringement by BN.

4.2 Graphical Interchange Format (GIF)

A second case concerns the patent award and alleged infringement pertaining to the LZW encoding method that underlies the .GIF document format, which is used widely for graphic images. This infringement suit was unsuccessful and eventually withdrawn.

GIF images are compressed to reduce file size, using a technique called LZW (after Lempel-Ziv-Welch). The technique was initially described by Welch in *IEEE Computer*, June 1984. Unisys holds a patent on the technique described in the article, but the article describing the algorithm made no mention of this. The LZW procedure quickly became a popular technique for data compression because it provided economical, high performance, adaptable, and reversible data compression. Likewise, GIF became a standard in its field. Apparently, neither CompuServe, the programmer who designed GIF, nor the software community were aware of the patent. Claim 1 of the patent reads as follows:

1. In a data compression and data decompression system, compression apparatus for compressing a stream of data character signals into a compressed stream of code signals, said compression apparatus comprising storage means for storing strings of data character signals encountered in said stream of data character signals, said stored strings having code signals associated therewith, respectively, means for searching said stream of data character signals by comparing said stream to said stored strings to determine the longest match therewith, means for inserting into said storage means, for storage therein, an extended string comprising said longest match with said stream of data character signals extended by the next data character signal following said longest match, means for assigning a code signal corresponding to said stored extended string, and means for providing the code signal associated with said longest match so as to provide said compressed stream of code signals.

CompuServe released GIF as a free and open specification in 1987. GIF soon became a world standard and played an important role in the Internet community. In December 1994, CompuServe and Unisys suddenly announced that developers would have to pay a license fee in order to use this technology that had been patented by Unisys. This caused immediate confusion.

The original licensing agreement text which had upset so many was soon followed by clarifications from CompuServe and Unisys. Unisys faced a delicate challenge and risked alienating a large segment of the software community. While having the right to file suit against LZW users, Unisys has been accommodating and fair. Most likely, the success of LZW and its widespread use caught Unisys off guard. It is difficult to understand how else Unisys could first allow a large number of developers to use LZW for free for years and then, after the establishment of de facto standards based on LZW, abruptly change its attitude.

5. DIRECTIVE ON THE PATENTABILITY OF COMPUTER- IMPLEMENTED INVENTIONS

The primary requirement of the Directive is that, “in order to be patentable, an invention that is implemented through the execution of software or on a computer, or similar apparatus has to make a contribution in a technical field that is not obvious to a person of normal skill in that field.” A computer-implemented invention is defined as “any invention implemented on a computer or similar apparatus which is realized by a computer program.” (Proposal for a Directive, 2002, p.13) Under the current situation, inventors have two avenues for obtaining protection for inventions. They may (1) apply for patents at the European Patent Office (EPO) under the auspices of the European Patent Convention, or (2) seek patents through the national patent offices in the Member States. Practically speaking, this scenario has resulted in differences of interpretation and claims that “there is no unifying structure with binding effect on national courts.” Thus, the Directive represents a call for harmonization. (http://www.europa.eu.int/comm/internal_market/en/indprop/comp/02-32.htm)

The formulation of the Directive has fueled a heated debate on the limits of what is patentable—and the benefits of software patenting—both currently and for the future. Detractors argue that the problems of strategic patenting are only now gaining attention in the U.S. and they warn that “software patents damage innovation by raising costs and uncertainties in assembling the many components needed for complex computer programs and constraining the speed and effectiveness of innovation” (Open letter, 2003). Clearly, however, some action is inevitable and required: “over the years, national courts have decided that there is no reason why a patent should not be granted for a machine programmed to carry out some technical function, or a technical process carried out using a computer or similar machine.” As a result, since the EPC came into force in 1978, more than 30,000 software patents have been granted and a body of case law has been built up by the EPO and Member States’ courts.

Those supporting the Directive, including Arlene McCarthy, JURI’s rapporteur and the European Parliament member responsible for the draft legislation, maintain that nothing will be made patentable which is not already patentable. Moreover, there are “links between the patentability of computer-related inventions and the growth of IT industries in the United States” (Broersma, August 27, 2003). Opponents to the Directive have responded that economic research does not support such links between software patents and business growth. They argue that patents are harmful in casting “in concrete the so-powerful oligopolies that naturally emerge in information-based industries” and that in the “United States, where such patents are allowed, large corporations such as IBM routinely stockpile patents to be used against competitors--usually to the detriment of smaller companies” (Broersma, April 28 & August 28, 2003). Others have long argued that patents can be used to delay or stifle innovation through the use of patent pools, patent thickets, exclusive licensing, and other abusive patent enforcement tactics (Rivette & Kline, 1999).

The authors of the Critique perceive significant uncertainties surround the scope, reach, and effect of software patents and limits on the ability of small companies to grow in competition with large companies. The Fall of

2003 saw demonstrations, protests, a call for action petition, and an open letter and critique written by a high ranking group of scientists. On September 15, 2003, the European Parliament voted to approve the Directive. However, this approval also included several amendments. The amendments outlaw patents for algorithms, make it impossible to register business-method patents, and restrict the definition of the sort of software that should be eligible for a patent. Supporters of software patents argue that the amendments could deprive existing patent holders of their ability to enforce their property rights and that such a law may violate international intellectual property agreements. Experts predict that the European Commission is likely to be displeased with the adjustments and that ministers of the 15 national governments of the union are likely to challenge the amendments to the Directive.

6. THE VALUE OF SOFTWARE PATENTS

Those who argue in favor of software patents identify five types of public benefits that can be realized:

Encouraging inventive efforts through the promise of economic rewards to inventors. Patents for new software are likely to encourage increased discovery and development of new types of useful software. Patent rewards can overcome “free rider” risks. Free riding slows development of new software because these efforts involve considerable financial outlays that may not produce accompanying returns if others can freely use the resulting product (Gruner, 2000, pp. 999-1007).

Promoting public disclosures of useful inventions through issued patents. In exchange for the public disclosure of an invention in a published patent, an inventor is given the reward of exclusive rights in the invention for 20 years. Disclosure makes the availability of the invention, under sale or license, known to those who may have a use for it. Disclosure encourages others to consider making improvements or other substitutes. Most important, public disclosure can avoid unnecessary duplication of efforts in creating something that already exists (Gruner, 2000, pp. 1007-12).

Providing incentives for product refinement and commercialization. Substantial post-invention refinement may be needed to begin mass production, distribution, and marketing of a new invention. Absent patent rights, a potential developer may be reluctant to be the first to take on the production problems and the marketing costs for a new product. Patent protection may be needed to convince investors to back start-up companies in the development and marketing of new software. Subsequent developers’ costs will be lower if they can gain some benefit from the first mover’s product introduction efforts (Gruner, 2000, pp. 1012-20).

Encouraging Efficient Prospecting for Applications and Improvements. Patents may encourage software developers to search for further applications of the patented software in the same field in which the software was originally developed, or promote cross-domain searching for new applications in other fields. (Gruner, 2000, pp.1019-23)

Limiting duplicative efforts to discover, perfect, and improve patented inventions, thereby maximizing society’s net gain from each patented invention. Software patents may prevent wasteful investments. Following a software developer’s patenting of a particular advance, others working to

develop similar or less effective software will be averted from further wasteful or expensive efforts. (Gruner, 2000, pp.1023-28)

Finally, we might ask whether alternative forms of intellectual property protection would provide the same public benefits. Those who argue that patents are preferable often compare them to copyright or trade secret protection. As to whether copyright protection is preferable to patent protection, we must remember that patents protect functional and utilitarian features of software that cannot be protected by copyright, which protects only descriptive or expressive aspects (Gruner, 2000, p. 994) In addition, unlike trade secret protection, patents provide rights that prevent reverse engineering and reuse of software (Gruner, 2000, p.995)

7. A REASSESSMENT

How is the European Union to proceed? While software patents may stymie innovation especially in small companies, and while the economic benefit is not resolved, some kind of harmonization appears necessary. Is the backlash alarmist given that “nothing will be made patentable which is not already patentable” or are the detractors wise to observe that this formalization of existing practice is dangerous?

We find limitations to the persuasiveness of the current arguments on both sides. Those who support the Directive warn of possible future conflicts in interpretation and application of patent law. But there is no hard evidence to corroborate this concern. Nor is there sufficient evidence and justification for whether the solution—the Directive as it is currently formulated—will remedy the problem. Those opposed to the Directive have taken an extreme, even Draconian position, arguing that this remedy will have dire consequences. They claim the Directive will spawn new problems for R&D and competition, framing slogans such as: “patent inflation is not a victimless crime” and “software patents kill innovation.”

Some have argued that patents on software may actually slow innovation by making it more costly and difficult to advance or build on existing information technology. This position draws on the “tragedy of the anticommons”, which theorizes that an over-assignment of property rights for a privately-held resource leads to under-utilization of that resource. Preventing an anti-commons tragedy “requires overcoming transaction costs, strategic behaviors, and cognitive biases of participants, with success more likely within close-knit communities than among hostile strangers” (Heller & Eisenberg, 1998, p. 280).

On the other hand, an example of where the cautionary reaction may be well founded concerns the patentability of pure business methods. Much controversy has surrounded this issue in the U.S. as to its effects on software development. In the EU, these will not be patentable; however, “some inventions involving business methods could fall within the definition of computer-implemented inventions. These inventions would be dealt with in accordance with the Directive, and in particular patents would only be granted for inventions that made a technical contribution” (Proposal FAQ p.5). Critics rightfully anticipate a natural progression of decisions that will follow, leading to more patents for business methods.

Critics have continued to argue for reforms to the patent system. For example, Gleick (2000) claims “that in the US, the patent office has grown entangled in philosophical confusion of its own making; it has become a

ferocious generator of litigation; and many technologists believe that it has begun to choke the very innovation it was meant to nourish” (Gleick, 2000).

One compromise solution would be to limit the term for software patents. Product life cycles in software are very short: approximately 3 years. What is seen as an invention at inception, can by virtue of adoption be mainstream within six months. On the other hand, a patent protects an invention for 20 years, which is clearly disproportionate. Likewise, Jeff Bezos of Amazon has suggested that a patent term of 3-5 years is more realistic and fair.

Moreover, the Directive has exposed ideological differences between two competing paradigms: Open Source versus pro-intellectual property rights. The Directive has alienated the Open Source community and the Euro Linux alliance—over 200 commercial software publishers and European non profit associations with the goal of promoting and protecting the use of Open Standards, Open Competition, and Open Source Software.

Regardless of the stated positions, the world is moving toward globalization and greater integration of trade, commerce, and intellectual property rights: “Finding the right balance will not be easy. Patents can be a spur to innovation, but they can also be an obstacle, and the great advantage of digital technology was supposed to be its very malleability. Moreover, there is another headache. The harder it is to patent computer-related inventions in Europe, the wider will be the legal gap with America” (Economist, 2003). In future, the U.S., EU, Japan, China, and the developing world will need to compromise and take steps in the direction of one another to find common ground.

Much remains to be understood about software patentability. Future directions for research might consider a range of empirical and economic issues. Specifically, we have yet to understand the relationship between the number of software patents awarded versus the number of patents litigated. We might discover, in fact, that most software patents are not the cause or source of litigation.

REFERENCES

- Battilana, Michael C. (2003). The GIF Controversy: A Software Developer's Perspective. <http://cloanto.com/users/mcb/19950127giflzw.html>
- Bessen, J.& Hunt, R.M. (2003). An empirical look at software patents, (*MIT and the Federal Reserve Bank of Philadelphia: working paper*, 2003, pp.7, 9, 31) <http://www.researchoninnovation.org/swpat.pdf>
- Broersma, Matthew (2003). Developers gather to protest patents. *CNET News.com*, Aug. 28, 2003. http://news.com.com/2102-1008_3-5069279.html?tag=ni_print
- Broersma, Matthew. (2003). EU rapped over software plan. *CNET News.com*, Aug. 27, 2003. http://news.com.com/2102-1008_3-5068842.html?tag=ni_print
- Broersma, Matthew (2003). Patent battle to culminate in Brussels. *CNET News.com*, August 26, 2003. http://news.com.com/2100-1012_3-5068007.html
- Broersma, Matthew (2003). Software patent vote delayed. *CNET News.com*, June 30, 2003. http://news.com.com/2100-1012_3-1022181.html?tag=rn
- Broersma, Matthew. (2003). Scientists protest EU software patents. *CNET News.com*, April 28, 2003. http://news.com.com/2102-1012_3-998547.html?tag=ni_print
- Burgunder, Lee. (2003). *Legal Aspects of Managing Technology*. Mason, Ohio: West Legal Studies in Business, pp.173-203.
- Computer-related Invention/VICOM*, T 208/84, 1987 OJ EPO 14 (1986).
- A Critique of the Rapporteur's Explanatory Statement* accompanying the JURI Report to the European Parliament on the proposed Directive on the Patentability of Computer-Implemented Inventions. (August 2003). Paul A. David, Oxford Internet Institute and Stanford University; Bronwyn H. Hall, University of California, Berkeley and Scuola Sant'anna Superiore Pisa; Brian Kahin, University of Michigan; W. Edward Steinmueller, Science and Technology Policy Research Unit, University of Sussex, August 2003 <http://www.researchineurope.org/policy/critique.htm>
- Diamond v. Diehr*, 450 U.S. 175 (1981).
- The Economist. (2003). A clicking bomb: An explosive row over how to protect intellectual property in Europe. September 4, 2003. http://www.economist.com/business/displayStory.cfm?story_id=2043416
- Examination Guidelines for Computer-Related Inventions*, (1998) U.S. Patent & Trademark Office. <http://www.uspto.gov>
- Final Report: The Results of the European Commission Consultation Exercise on the Patentability of Computer Implemented Inventions (PbT Consultants). http://www.europa.eu.int/comm/internal_market/en/indprop/comp/studyintro.htm
- Dinwoodie, Graeme B., Hennessey, William O., & Perlmutter, Shira. (2002). *International and comparative patent law*. Newark, New Jersey: LexisNexis Group, pp. 89-115.
- Gleick, James. (2000). Patently absurd. *New York Times Magazine*, March 12, 2000. <http://www.nytimes.com/library/magazine/home/20000312mag-patents.html>
- Gratton, Eloise, (2003). Should patent protection be considered for computer-related innovations? *Computer Law Review & Technology Journal* 7, 223-253.
- Gruner, Richard S.(2000). Better living through software: Promoting information processing advances through patent incentives, *St. John's Law Review* 74, 977-1068. *Guidelines for computer-software related inventions*, (2000). Japanese Patent Office. <http://www.jpo.go.jp>
- Heller, Michael A. (1998) The tragedy of the anticommons: Property in the transition from Marx to markets, *Harvard Law Review* 111, 621-688.
- Heller, Michael, A. & Eisenberg, Rebecca S. (1998) Can patents deter innovation? The anticommons in Biomedical Research, *Science* 280 (5364), 698-701.
- Japanese Patent Law, articles 2 & 29. <http://www.jpo.go.jp>
- League of Programming Freedom. <http://www.lpf.ai.mit.edu/>
- An Open Letter to the European Parliament* Concerning the Proposed Directive on the Patentability of Computer-Implemented Inventions (25 August 2003). Birgitte Andersen, Birkbeck, University of London; Paul A. David, Oxford Internet Institute and Stanford University; Lee N. Davis, Copenhagen Business School; Giovanni Dosi, Scuola Sant'anna Superiore; David Encaoua, Université Paris I; Dominique Foray, IMRI Université Dauphine; Alfonso Gambardella, Scuola Sant'anna Superiore; Aldo Geuna, SPRU, University of Sussex; Bronwyn H. Hall, University of California, Berkeley and Scuola Sant'anna Superiore; Dietmar Harhoff, Ludwig-Maxmiliens Universitaet; Peter Holmes, SEI, University of Sussex; Luc Soete, MERIT, University of Maastricht; W. Edward Steinmueller, SPRU, University of Sussex.

http://www.researchineurope.org/policy/patentdirltr.htm#_ftn1

Pension Benefit Systems, 2001 OJ EPO 413 (2001).

Proposal for a Directive of the European Parliament and of the Council on the Patentability of Computer-Implemented Inventions. Brussels, 20.02.2002. COM(2002)92 final. 2002/0047 (COD)

Proposal for a Directive of the European Parliament and of the Council on the Patentability of Computer-Implemented Inventions-frequently asked questions. (FAQ)

http://www.europa.eu.int/comm/internal_market/en/indprop/comp/02-32.htm

Rivette, Kevin & Kline, David. (1999). *Rembrandts in the attic: Unlocking the hidden value of patents*. Cambridge, Mass: Harvard Business School Publishing.

Software Patent Institute - <http://www.spi.org/primintr.htm>,

State Street Bank & Trust Co. v. Signature Financial Group, Inc., 149 F.3d 1368 (Fed. Cir. 1998).

Tocups, Nora M., & O'Connell, Robert J. (1997). Patent protection for computer software. *Computer Law* 14(11), 18-23.

United States Code, title 35, sections 101, 102, 103, & 271.

APPENDIX

Number of Software Patents Granted in the U.S.*

Year	Software Patents	Aharonian Estimates	Other Utility Patents	Software/Total
1976	766	100	69,460	1.1%
1977	885	100	64,384	1.4%
1978	902	150	65,200	1.4%
1979	800	200	48,054	1.6%
1980	1,080	250	60,739	1.7%
1981	1,281	300	64,490	1.9%
1982	1,404	300	56,484	2.4%
1983	1,444	350	55,416	2.5%
1984	1,941	400	65,259	2.9%
1985	2,460	500	69,201	3.4%
1986	2,666	600	68,194	3.8%
1987	3,549	800	79,403	4.3%
1988	3,507	800	74,417	4.5%
1989	5,002	1,600	90,535	5.2%
1990	4,738	1,300	85,626	5.2%
1991	5,401	1,500	91,112	5.6%
1992	5,938	1,624	91,506	6.1%
1993	6,902	2,400	91,440	7.0%
1994	8,183	4,569	93,493	8.0%
1995	9,186	6,142	92,233	9.1%
1996	11,664	9,000	97,981	10.6%
1997	12,810	13,000	99,173	11.4%
1998	20,411	17,500	127,108	13.8%
1999	21,770	21,000	131,716	14.2%
2000	23,141	--	134,454	14.7%
2001	25,973	--	140,185	15.6%

*Source: J. Bessen & R. M. Hunt, "An Empirical Look at Software Patents," p. 31 (MIT and the Federal Reserve Bank of Philadelphia: working paper, 2003) <http://www.researchoninnovation.org/swpat.pdf>