GaTech TI:GER Working Paper Series

Software Patents: Good News or Bad News?

Stuart J.H. Graham
College of Management, Georgia Institute of Technology

David C. Mowery
Haas School of Business, UC Berkeley, Harvard Business School, and NBER

May 2004

PRELIMINARY DRAFT: DO NOT CITE OR QUOTE WITHOUT AUTHORS' WRITTEN PERMISSION

An earlier version of this paper was presented at the AEI-Brookings Joint Center conference on "Intellectual Property Rights in Frontier Industries," American Enterprise Institute, Washington, D.C., April 30, 2004. We appreciate comments from conference participants. Research for this paper was supported by the Alfred P. Sloan and Andrew W. Mellon Foundations.

I. Introduction

Although measures and definitions of "software patents" vary, virtually all analyses (including the data presented below) agree that patenting in this field has grown since the early 1980s. Controversy over the validity and economic consequences of software patents also has grown during this period. Software patents have been criticized for their poor quality, for their chilling effects on innovation, and for reducing R&D spending in the software industry. In this paper, we review some of these controversies, emphasizing the paucity of strong evidence to support these contentions. Evidence also is lacking, however, in support of arguments that patents are essential to innovation and technological progress in computer software, a technological field in which innovation proceeded apace for at least 30 years before patenting began to grow rapidly. Indeed, it is difficult to understand the causes of recent growth in software patenting without a brief historical review of the development of the U.S. software industry and (of no less importance) the development of judicial decisions on the validity and coverage of patents and other legal devices for protecting software-related intellectual property. Nevertheless, the growth of software patenting highlights the U.S. patent system's difficulties in dealing with new areas of patenting, especially when growth of a new field of patenting flows from changes in the legal strength of patents and industry structure rather than from the development of a new field of technology.

Immediately below, we discuss the general issues and evidence (or lack of same) relating to the role of patents in technological innovation. This section is followed by a brief survey of the development of the U.S. computer software industry and the judicial decisions on the validity of patents and other legal forms of intellectual property protection for software during the 1980s and 1990s. We then discuss our data on trends in patenting, focusing in particular on packaged software. A discussion of the complex issues of patent "quality" is followed by a conclusion and consideration of policy implications.

Our basic conclusions can be summarized as follows: Although patents and other forms of legal protection for intellectual property have assumed greater importance in the software industry since the early 1980s, the long history of innovation in the sector makes it difficult to argue that strong patent protection is indispensable for software innovation. And it is possible that various forms of "strategic patenting" could impede innovation in this technology. Nevertheless, strong evidence of such impediments is lacking. Moreover, the appropriate policy response to a software patenting "problem" is not obvious—*sui generis* protection, for instance, may create as many problems as it resolves. The growth of software patenting and software patenting controversies, however, raise broader issues concerning the weaknesses of the "quality control" procedures within the US Patent and Trademark Office's administrative reviews of patent applications. These procedural weaknesses affect patenting across the board, although their effects are likely to be most significant in new technological fields for which patent-based prior art is relatively scarce and Patent Office technical expertise may be lacking.

II. Patents and innovation

In order to highlight some of the key policy issues in any discussion of patenting in software, we begin with a short review of recent work on the links between patents and innovation. Despite the recent upsurge in research on the economics of innovation and intellectual property, surprisingly little new empirical research provides answers to such critical issues as the importance of patents for innovation. Instead, the evidence is mixed and ambiguous at best.

Mazzoleni and Nelson (1998) provide an excellent overview of the economic evidence concerning the roles of patents in innovation, and argue that economists have highlighted four key functions of the patent system in the innovation process:

(1) providing incentives for inventive activity, by enabling inventors to capture the returns to their investments of funds and time in inventive activity;

(2) providing incentives for the commercial development of inventions, by enabling investors in commercialization to capture the returns to their investments in technology development and supporting markets for well-defined pieces of intellectual property;

(3) providing incentives for inventors to disclose technical details of their inventions through publication of patents; and

(4) enabling the orderly development of "follow-on" inventions.

The authors note that these arguments have figured prominently in the series of U.S. policy initiatives dating back to the 1980s that have strengthened patentholder rights in domestic and international markets.

But Mazzoleni and Nelson also emphasize the weak evidentiary basis for strong claims about the contributions of patents to invention and innovation. Surveys and other empirical work (Mansfield, 1986; Levin et al. 1987; Cohen et al., 2001), for example, suggest that the "invention incentive" offered by patents is limited in most industries other than pharmaceuticals and scientific instruments—other mechanisms or strategies are typically rated more highly by R&D managers as means of capturing the returns to invention. Little evidence supports the argument that patents provide strong support for technological development, although this premise influenced the passage of important pieces of legislation such as the Bayh-Dole Act of 1980 (See Mowery et al., 2004, for a review of the Bayh-Dole Act's origins and effects). Clearly, there are examples of industries (e.g., biotechnology) in which stronger patents have supported the growth of markets for intellectual property that have in turn created a vertically specialized industry structure. But there is not much evidence that the vertically specialized industry structure now characteristic of computer hardware and software owes much to the development of strong intellectual property rights. The "disclosure effects" of patent publication have been less closely examined, as have those of the broad patents deemed most helpful to commercial exploitation of "prospects" for the creation of a series of inventions. Merges and Nelson (1990), however, argue that broad patents may have a chilling effect on entry, blocking other inventors from the exploitation of broad technological opportunities. And Scotchmer and Green (1990) suggest that difficulties in developing efficient licensing arrangements for such broad patents may impede follow-on inventive activity by parties other than the original inventor. Overall, however, the evidence here too is ambiguous.

Another issue that figures prominently in any discussion of software patents is patent "quality"—i.e., are patents being issued for inventions that genuinely meet the criteria of non-obviousness and utility, or do weak procedures for review of patent applications result in the issue of patents whose quality is dubious or uncertain, pending a court challenge? As we note below, the processes through which U.S. patent applications are reviewed necessarily are backward-looking, since they focus on prior art, much of which is embodied in previously issued patents. New fields of inventive activity therefore pose significant challenges to patent review procedures. Some of these problems can be overcome through procedures that enable “interested parties” to challenge patents immediately after issue through administrative procedures, such as the “opposition proceedings” of the European Patent Office and numerous European nations. But in the absence of well-developed administrative procedures allowing challenges to patent validity, litigation provides a costly and time-consuming alternative that may prolong uncertainty over patent validity and have a chilling effect on incentives to invent, markets for intellectual property, and investment in “follow-on” inventions.

The links between patenting and innovation thus remain hazy, although little if any of this research has included software within the sample of industries or technologies examined. Our discussion below of trends in software patenting and innovative activity during the 1978-2003 period reveals little evidence that increased software patenting reflects much more than the response of managers to the greater ease of obtaining software patents. Moreover, as we pointed out earlier, vertical specialization in this industry predates the era of stronger patent protection. At the same time, there is little evidence that stronger patents have had a chilling effect on innovation in this field, nor do we find strong evidence that “follow-on” invention is being impeded by increased fragmentation or complexity in the intellectual property rights environment. Nor do we find that the “quality” of software patents has declined precipitously during this period, although the relatively weak procedures within the US patent system for administrative challenges to patent validity mean that the problem of “junk patents” should not be minimized.

III. The evolution of intellectual property rights policy and practice in the US software industry

Reliance by innovators on patents and other forms of formal intellectual property (IP) protection has changed greatly during the 50-year history of the U.S. computer software industry. These shifts in the role of formal IP protection reflect changes in the structure of the software industry and the markets for its products, as well as important developments in U.S. policy and judicial interpretation of patents and copyright in the field of software. Although the bulk of our discussion focuses on patents in software, we include a brief overview of developments in software copyright protection since judicial decisions on the scope of software copyrights have affected innovators' incentives to seek patents. In addition, of course, software innovators have relied on alternative methods, such as trade secrets,[1] legal assertions of misappropriation,[2] trademark,[3] and even the Semiconductor Chip Protection Act,[4] to protect their intellectual property.

A. Evolution of the computer software industry's structure

For the first 25 years of the computer industry's history, computer software was rarely sold as a separately priced and marketed product. Beginning in the late 1960s, however, producers of mainframe computers "unbundled" their software product offerings from their hardware products, separating the pricing and distribution of hardware and software. This development provided opportunities for entry by

1 A trade secret is formally some information used in a business which, when secret, gives one an advantage over competitors. The secret must be both novel and valuable. Metallurgical Industries, Inc. v. Fourtek, Inc. 790 F.2d 1195 (1986).
2 Collectors of valuable information can prevent competitors from using the information. International News Service v. Associated Press, 248 U.S. 215 (1911).
3 Protects names, words, and symbols used to identify or distinguish goods and to identify the producer. Zatrains, Inc. v. Oak Grove Smokehouse, Inc. 698 F.2d 786 (5th Cir., 1983).
4 Protection is available for software embodied in semiconductor chips—so-called mask works. E.F. Johnson v. Uniden Corp. of America, 653 F. Supp. 1485 (D. Minn. 1985).

independent producers of standard and custom operating systems, as well as independent suppliers of applications software for mainframes.

Rapid adoption of the desktop computer in the United States after 1980 supported the early emergence of a few “dominant designs” in desktop computer architecture, further eroded vertical integration between hardware and software producers, and opened up opportunities for ISVs to enter the first mass market for packaged software. Few of the major suppliers of desktop software came from the ranks of the leading independent producers of mainframe and minicomputer software, and mainframe and minicomputer ISVs are still minor factors in desktop software. Interestingly, however, vertical specialization emerged during a period of relatively weak formal protection for software-related intellectual property. The packaged computer software industry now has a cost structure that resembles that of the publishing and entertainment industries much more than that of custom software--the returns to a “hit” product are enormous and production costs are low. And like these other industries, the growth of a mass market for software has elevated the importance of formal intellectual property rights.

Since 1994, the development of the software industry has been dominated by the growth of networking among desktop computers within enterprises through local area networks linked to a server and/or the Internet, which links millions of users. How has the growth of the Internet changed the economics of intellectual property protection in the software industry? At least three different effects are apparent thus far in the Internet’s development. First, the widespread diffusion of the Internet has created new channels for low-cost distribution and marketing of packaged software, reducing the barriers to entry into the packaged software industry that are based on the dominance of established distribution channels by large packaged software firms. In this respect, the Internet expands the possibilities for rapid penetration of markets by a "hit" packaged software product (in the jargon of the software industry, a "killer app"), which enhances the economic importance of protection for these types of intellectual property. The Internet also is an important factor in the growth of patents on software-embodied "business methods," many of which concern tools or routines employed by on-line marketers of goods and services.

But the Internet also has provided new impetus to the diffusion and rapid growth of a very different type of software, "open source" software. Although so-called "shareware" has been important throughout the development of the software industry, the Internet's ability to support rapid, low-cost distribution of new software and (crucially) the centralized collection and incorporation into that software of improvements from users has made possible such widely used operating systems as Linux and Apache (See Kuan, 1999 and Lerner and Tirole, 2000). The Internet thus has increased the importance of formal protection of some types of software-related intellectual property, while simultaneously supporting the growth of open source software, which does not rely on such formal instruments of IP protection (Graham and Mowery, 2004).

B. Software Copyright Protection

Copyright protection for software innovation was singled out by policymakers during the 1970s as the preferred means for protecting software-related intellectual property (Menell, 1989). In its 1979 report, the National Commission on New Technological Uses of Copyrighted Works (CONTU), charged with making recommendations to Congress regarding software protection, chose copyright as the most appropriate form of protection for computer software (CONTU, 1979). Congress adopted the Commission's position when it wrote "computer program" into the Copyright Act in 1980.[5]

The federal judiciary's application of copyright law to software in the aftermath of the CONTU initially promised strong protection for inventors. *Apple Computer, Inc. v. Franklin Computer Corp*.[6] is an early and important case of copyright litigation in packaged software. Although the federal judiciary had long held that copyright protected only "expression" in works,[7] the Court in *Apple Computer* held that Apple's code was protected by its copyright. This decision strengthened copyright protection considerably, making it possible for one firm's copyrighted software to block the innovative efforts of others. Other decisions—the so-called "look and feel" cases—extended traditional copyright protection of "expression" to such "non-literal" elements of software as structure, sequence, and organization.[8]

Subsequent court decisions, however, narrowed the protection provided by copyright for software-related intellectual property. The sweeping interpretation of copyright protection in the *Apple Computer* case was narrowed and weakened in a series of copyright infringement cases brought by Lotus Development. The *Borland* decision weakened the strong protection for software inventions provided by *Apple Computer v. Franklin Computer*, and along with other decisions affirming the strength of software patents, may have contributed to increased reliance by some U.S. software firms on patents in the 1990s.

---

5 17 U.S.C. sec. 101, sec. 117 (as amended 1980). For a more complete discussion, see Menell (1989).
6 714 F.2d 1240 (3rd Cir. 1983). Consistent with its position as a leading firm in the packaged software industry Microsoft, which supported stronger formal protection for software-related intellectual property, filed an *amicus curiae* brief on behalf of Apple in this case.
7 Historically, a major distinction in the copyright law has been that ideas are not protected, only expressions are. Baker v. Selden, 101 U.S. 99 (1879).
8 Computer Associates Int'l v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992); Whelan Associates v. Jaslow Dental Laboratory, 797 F.2d 1222 (3rd Cir. 1986).

C. Patents in Software

In contrast to copyright, federal court decisions since 1980 have broadened and strengthened the economic value of software patents. In the cases of *Diamond v. Diehr*[9] and *Diamond v. Bradley,*[10] both decided in 1981, the Supreme Court announced a more liberal rule that permitted the patenting of software algorithms, strengthening patent protection for software (Merges, 1996). The economic value of these patents was highlighted in several high-profile cases during the 1990s. For example, a 1994 court decision found Microsoft liable for patent infringement and awarded $120 million in damages to Stac Electronics. The damages award was hardly a crippling blow to Microsoft, but the firm's infringing product had to be withdrawn from the market temporarily, compounding the financial and commercial consequences of the decision (Merges, 1996).

As the USPTO adopted a more favorable posture toward applications for software patents, the quality of issued patents in an area of technology in which patents historically had not been used to cover major innovations was called into question. The celebrated "multimedia" patent issued by the USPTO to Compton Encyclopedias in 1993 is one example of these difficulties. On November 15, 1993, Compton's Newmedia announced that it had won a "fundamental" patent for its multimedia software that rapidly fetched images and sound (Peltz, 1993; Markoff, 1993). The patent was very broad, and the firm's president sought to negotiate licensing agreements with competitors, in order to "…promote a standard that can be used by every developer, and be compensated for the investments we have made."[11] On March 28, 1994, the USPTO released a preliminary

---

9 450 U.S. 175 (1981).
10 450 U.S. 381 (1981).
11 Abate, T. "Smaller, faster, better; Tech firms show off their latest wonders at trade show and foretell a

statement declaring that "[a]ll claims in Compton's multimedia patent issued in August 1993 have been rejected on the grounds that they lack 'novelty' or are obvious in view of prior art."[12] This declaration was confirmed by the Office in November of 1994.[13]

Similar concerns over the validity or quality of software-related patents appeared in the field of "business methods" patents, a field of patenting that grew rapidly in the wake of the 1998 decision of the Court of Appeals for the Federal Circuit (CAFC) affirming the validity of a "business methods" software patent in *State Street Bank v. Signature Financial Group*.[14]

USPTO Commissioner Dickinson noted in March 2000 that the number of applications for such patents had expanded from 1,275 in fiscal 1998 to 2,600 in fiscal 1999, resulting in the issue of 600 business methods patents in 1999.

Political reactions to the surge in business methods patents and the controversy surrounding their validity were swift and involved both Congress and the USPTO. In late 1999, Congress passed the "American Inventor Protection Act" (AIPA), which included a "first-to-invent" defense against infringement claims that protects defendants who can show that they were practicing the relevant method or art one year or more prior to the filing of the patent application against infringement suits.[15] The USPTO's "Business Methods Patent Initiative," unveiled in the spring of 2000, expanded the number of USPTO examiners charged with reviewing business-method patent applications and introduced a "second review" for all such applications.

---

user-friendly future," San Francisco Examiner, November 21, 1993, E:1.

12 Riordan, T. "Action Was Preliminary On a Disputed Patent," The New York Times, March 30, 1994, D:7.

13 Orenstein, S. "U.S. Rejects Multimedia Patent," The Recorder, November 1, 1994, 4.

14 149 F.3d 1368 (CAFC, 1998).

15 The AIPA was originally drafted to revise the U.S. patent system to be consistent with the WTO agreements that concluded the Uruguay Round of trade negotiations, but additional provisions were added specifically to address the business methods patent controversy.

This administrative initiative appears to have reduced the rate of issue of new patents in this class. The USPTO reported in 2001 that the number of examiners assigned to business methods patents increased from 45 at the beginning of fiscal 2000 to 82 by the end of fiscal 2001. The same report predicted that roughly 10,000 applications would be filed in "Class 705," which covers most business methods patents, in fiscal 2001, an increase of nearly fourfold since fiscal 1999. But the USPTO issued approximately 433 patents in Class 705 in fiscal 2001, a decrease of more than 25% from the number issued in this class in fiscal 1999.[16]

The economic significance and validity of U.S. business methods patents ultimately will be determined through litigation.[17] The controversies over the Compton's patent and business methods patents are not confined to software, however, and reflect the problems that result in any new field of patenting, because of the reliance by USPTO examiners on patent-based prior art in their reviews of applications.[18] As we note below, administrative procedures for challenging the validity of patents in the U.S. patent system are weak, leaving litigation, which is both costly and time-consuming, as the only alternative.

---

16 See http://www.uspto.gov/web/menu/pbmethod/fy2001strport.html. According to the USPTO "Cassis" database, 449 and 423 patents were issued in Class 705 respectively for 2002 and 2003.

17 The Internet vendor of books and other products, Amazon.com, filed suit in 1999 against Barnes & Noble over the latter's alleged infringement of its patent on "one click" order methods. Although Amazon was granted an injunction against Barnes & Noble's alleged infringement of its "one-click" patent by the District Court for the Western District of Washington State in December 1999, the Court of Appeals reversed the judge in February 2001 and remanded the case to the district court. Given the CAFC's central role in establishing the patentability of business methods, its reversal of an injunction in this case is noteworthy.

18 "Now we're dealing with a much broader universe of 'prior art,' says J.T. Westermeier, a Washington D.C. internet attorney with Piper and Marbury, pointing out that many allegedly novel Internet business methods may already have been in use at universities or elsewhere." (Waldmeir and Kehoe, 1999).

IV. Patenting trends in the U.S. software industry, 1987-2003

This section examines trends in U.S. software patenting during the 1987-2003, focusing on the product areas that we believe have been most affected by changes in the legal treatment of software patents—packaged software. As our previous paper (Graham and Mowery, 2003) emphasized, no widely accepted definition of "software patent" exists, and the problem of identifying software patents is not made easier by USPTO changes in patent classification schemes. Other researchers have chosen to define a "software patent" by reference to certain key words in the patent disclosure (Bessen and Hunt, 2004), but we rely instead on the classification decisions of USPTO patent examiners.

One difficulty that arises when using patent office classifications is the rapid growth in the number of software-related USPTO patents during the period of this analysis. Because we are interested in analyzing changes over time in the number of software patents, we seek to insulate our sample from any "reclassifications" of patents from "all other" to a "software-related" category. Such an analysis is made more difficult by the USPTO's unannounced reclassifications. In order to eliminate the impact of reclassifications, we use a "snapshot" in time of the US classifications for all US-issued patents.

Using the USPTO's "Cassis" database, we collected data on all patents issued in the U.S. 1987-2003, relying on the USPTO classification scheme as of December 2003, with patents issued in previous years updated to reflect current USPTO classifications and thus allowing us to compare trends in patenting over time. We concentrated on the following USPTO classifications:

Class 345: COMPUTER GRAPHICS PROCESSING, OPERATOR INTERFACE PROCESSING, AND SELECTIVE VISUAL DISPLAY SYSTEMS
Class 358: FACSIMILE AND STATIC PRESENTATION PROCESSING
Class 382: IMAGE ANALYSIS
Class 704: DATA PROCESSING: SPEECH SIGNAL PROCESSING, LINGUISTICS, LANGUAGE TRANSLATION, AND AUDIO COMPRESSION/DECOMPRESSION
Class 707: DATA PROCESSING: DATABASE AND FILE MANAGEMENT OR DATA STRUCTURES
Class 709: ELECTRICAL COMPUTERS AND DIGITAL PROCESSING SYSTEMS: MULTIPLE COMPUTER OR PROCESS Class 715 DATA PROCESSING: PRESENTATION PROCESSING OF DOCUMENT
Class 710: ELECTRICAL COMPUTERS AND DIGITAL DATA PROCESSING SYSTEMS: INPUT/OUTPUT
Class 711: ELECTRICAL COMPUTERS AND DIGITAL PROCESSING SYSTEMS: MEMORY
Class 713: ELECTRICAL COMPUTERS AND DIGITAL PROCESSING SYSTEMS: SUPPORT
Class 714: ERROR DETECTION/CORRECTION AND FAULT DETECTION/RECOVERY
Class 717: DATA PROCESSING: SOFTWARE DEVELOPMENT, INSTALLATION, AND MANAGEMENT
Class 719: ELECTRICAL COMPUTERS AND DIGITAL PROCESSING SYSTEMS: INTERPROGRAM COMMUNICATION OR INTERPROCESS COMMUNICATION (IPC)

These classifications were identified by examining U.S. patenting in the years 1987-2003 by the six largest U.S. producers of personal computer software, based on their calendar 2000 revenues.[19] These patent classes account for 65.7% of the more than 3,800 patents assigned to the 100 largest packaged-software firms identified by Softletter, a trade newsletter, in its 2001 tabulation.[20] These 12 classes account for 67.9% of the

19 As reported in the Softletter 100 (2001) this group includes Microsoft, Novell, Adobe Systems, Autodesk, Intuit, and Symantec. We chose to focus our analysis on the patents assigned to specialized, publicly traded software firms, because we want to make use of R&D spending figures to compute a "software patent propensity" measure (software patents deflated by R&D spending): This measure is meaningful only for firms reporting R&D spending for which one can assume that the bulk of this R&D spending is devoted to software development, thus precluding many more generalized electronics firms (although these latter firms are engaged in a substantial amount of "software" patenting).
20 We are grateful to Softletter for permission to use these data. The 2001 tabulation was the last year to date in which Softletter produced this report.

patenting of the "top 6" firms, while these same 6 firms account for 88.4% of the patenting of the Softletter "top 100." Table 1 contains data on the distribution of the patents in this sample among the 12 USPTO classes listed above.[21]

Table 1: Patenting by the Softletter 100 (2001), by USPTO patent class, 1987-2003 (total patents=3,891)

U.S. Patent Class	Patent Count	Share of All Firm Patents	Cumulative Total
345	730	18.8%	18.8%
707	624	16.0%	34.8%
709	363	9.3%	44.1%
382	157	4.0%	48.2%
713	141	3.6%	51.8%
704	125	3.2%	55.0%
717	118	3.0%	58.0%
714	85	2.2%	60.2%
711	80	2.1%	62.3%
710	59	1.5%	63.8%
358	52	1.3%	65.1%
715	25	0.6%	65.8%

Our definitional scheme does not cover all software patents, but it does provide longitudinal coverage of a particularly dynamic and important segment of the overall software industry, inasmuch as the global market for packaged software represents some 50% of the total software and software-services industry output.[22] The data in Figure 1 indicate that the share of all U.S. patents accounted for by software patents grew from 2.1% to 7.4% of all issued US patents between 1987 and 1998, and the share of patents in these 12 US classes has remained between 6.9 and 7.5% of overall patenting during

21 We also constructed a sample of software patents that was based on the International Patent Classes associated with patenting by large packaged software specialists during this period of time. Overall, the IPC-based definition yields similar results and trends, but space constraints prevent a fuller discussion of this issue.
22 "National Software Strategy for Scotland," Scottish Enterprise and the Scottish Software Federation, 2001.

1999-2003. This slowdown in the rate of growth in "software" patenting as a share of total US patenting occurs in virtually every USPTO class included in our definition of software patents.[23]

FIGURE 1 HERE

There are several potential explanations for the slowdown in software patent growth, relative to overall US patenting, after 1999. In other work (Graham and Mowery, 2004), we noted that 1995 changes in the legal patent term of protection (changing the term of protection from 17 years from date of application to 20 years from date of issue) created strong incentives for patent applicants to pursue "continuations" in their applications, which (among other advantages) enabled applicants to extend the period of secrecy for their patent applications. Since software-patent applicants made extensive use of continuations, it is possible that the 1995 changes in patent term (whose effects on applicants' incentives to pursue continuations have been intensified by more recent requirements to publish many patents within 18 months of application) may have reduced incentives for software inventors to seek patents. It is also possible, however, that the accumulation of experience by USPTO examiners in dealing with software-patent applications, as well as the expanding body of patent-based prior art on which examiners rely in part, have led to lower rates of issue for software patent applications. The fluctuations in growth in software patents do not appear to be associated with fluctuations in the "pendency" of patent applications (the length of time required to review and grant or deny patent applications), since the average pendency of applications

23 The only class that did not show a reasonably flat growth trajectory after 1998 was the IPC group "G06F 17" which is the newest "software" class, and showed steady growth as a share of all patenting 1995-2003.

for issued software patents, which is greater than the average for all patents, has increased steadily through the 1995-2003 period.[24]

A. Software-related patenting by packaged software and electronic systems firms, 1987-2003.

In this section, we analyze patenting by U.S. software firms during 1987-2003, focusing on leading U.S. packaged software firms identified by Softletter in their 2001 tabulation of the 100 largest U.S. packaged software firms (based on revenues). We are particularly interested in the behavior of firms' "patent propensity" (patents per R&D dollar) over time. Since the R&D spending of these firms is devoted primarily to software development, in contrast to the software patenting of electronic-systems firms that do not separately report software-related R&D investment, a meaningful software-specific "patent propensity"can be computed only for packaged-software firms.[25] We also focus on these firms because the changes in the legal and policy environment discussed earlier may have had a more substantial effect on these companies.

Figure 2 displays trends during 1987-2003 in the share of all U.S. software patents held by the 100 largest U.S. packaged software firms, comparing trends that include and exclude the largest player in the industry, Microsoft. Figure 2 demonstrates that these firms increased their share of overall software patenting during the 1987-2003 period, from less than .06% in of all software patents in 1988 to nearly 4.75% in 2002, declining to 4.13% of software patents in 2003. Eliminating Microsoft from the Figure reveals more modest growth, with shares growing from less than .06% in 1987 to 1.35%

24 Pendency for software applications at issue for all software patents compared to all non-software patents from 1995-2003 are:

Year	1995	1996	1997	1998	1999	2000	2001	2002	2003
Pendency	130%	127%	122%	120%	122%	128%	136%	145%	146%

25 IBM does report firmwide software-specific R&D investment, and we examine this firm's software patent propensity in a later section.

in 2000 and declining to 1.0% in 2003. Similarly to Figure 1, the data in Figure 2 suggest rapid growth in software patenting through the late 1990s, followed by no growth or declines after 2000.

FIGURE 2 HERE

Although patenting by large packaged-software firms has grown since the late 1980s, it is interesting and surprising to note that electronic systems firms account for a larger share of software patenting as we define it (a definition that weights packaged-software patents more heavily than the schema developed by Bessen and Hunt, 2004). Both our USPTO and IPC classification schemes show that the share of overall "software" patents accounted for by large electronic systems firms (IBM, Intel, Hewlett-Packard, Motorola, National Semiconductor, NEC, Digital Equipment Corporation, Compaq, Hitachi, Fujitsu, Texas Instruments, and Toshiba) considerably exceeds the share of "software" patents assigned to specialist packaged-software firms. Figure 3 depicts the share of "software" patents assigned to our sample of 12 "electronics systems" firms, which fluctuates between a low of 21% in 1990 and a high of 28% in 1994 before falling to 21-23% of all software patenting for 1998-2003.[26]

FIGURE 3 HERE

We also calculate the share of all patents issued to these firms that we classify as "software" patents during 1987-2003. Figure 4 displays the time trend for the share of these patents within these 12 firms' patent portfolios during 1987-2003. Software patents' share of overall firm patents increases during the 1987-2003 period for all of these firms, from roughly 14% in 1987 to 25% of their overall patent portfolios by 2003.

---

26 Data analysis using the IPC-definition method found substantially the same trend in these firms' patenting shares.

Even more striking, however, is the level and growth of software patenting by IBM, the largest U.S. patenter,[27] which increased its software patenting from 27% of its overall patenting in 1987 to 42% in 2003. In contrast to the software patenting of the other eleven systems firms, IBM's share of "software patents" in its annual patenting increases through 2003.

FIGURE 4 HERE

B. Change in the "patent propensity" of packaged-software firms, 1987-2003

The data we have presented thus far suggest that software patenting has grown during at least the first part of the 1987-2003 period, but provide no information on the "intensity" of patenting effort by the firms obtaining software patents. Obviously, the interpretation of an increase in software patenting that is proportionate to growth in R&D inputs will differ from a finding that patenting is growing more rapidly than R&D investment. The first case may constitute evidence of intensified innovative effort in software in response to stronger patents. The second case suggests that stronger patents lead to more patenting relative to R&D investment. Accordingly, we examine the relationship between patenting and software-related R&D investment for the firms for which the latter data are available (unfortunately, we lack these data for most of our electronic systems firms) during 1987-2002.

The list of packaged-software firms provided by Softletter allow us to analyze the large packaged-software firms' "propensity to patent," measured as the ratio of patents to constant-dollar R&D spending, during the 1987-2002 period (Figures 5-6).[28] Figure 5

---

27 The USPTO reports in "Patenting by Organizations: 2001" (http://www.uspto.gov/go/taf/topo_01.pdf) that IBM (International Business Machine) is the largest patenting organization in the U.S., having been assigned 2,886 and 3,411 patents in 2000 and 2001, respectively. These numbers were 43% and 75% larger, respectively, than the next largest patenter in these years (NEC).

28 This analysis covers only the 1987-2002 period because corporate R&D investment data for 2003 were not available for all of the Softletter firms.

displays data on trends in the weighted-average patent propensity for the 15 largest U.S.-based packaged software firms.[29] These firms include the largest 15 publicly-traded firms by year 2000 revenue, as follows: Microsoft, Adobe Systems, Novell, Intuit, Autodesk, Symantec, Network Associates, Citrix Systems, Macromedia, Great Plains Software, RealNetworks, Borland (Inprise), SPSS, Phoenix Technologies, and SCO (Santa Cruz Operation). Figure 5 also compares the patent propensity of Microsoft with the aggregate propensity of the other 14 largest software firms (Microsoft is by far the largest patenter among the group, accounting for more than 75% of these firms' U.S. patents issued during 2000-2003).

Figure 5 demonstrates that patenting per R&D dollar has grown during the 1987-2002 period, in spite of a downturn during 2000-2001. This pattern is apparent both in the patent propensity of all 15 firms and in the sample of firms that excludes Microsoft. The Figure suggests that the "propensity to patent" of these firms grew especially rapidly during 1994-2000, a development that may reflect the policy changes (discussed above) affecting patent term that came into effect in the U.S. in 1995. As we noted above, changes in the patent term that became effective in 1995 created incentives for applicants to "rush" applications to the patent office prior to the 1995 change, a phenomenon reflected in the "bulge" in patents issued to these firms during 1994-2000. Interestingly, the sharp drops in R&D investment reported by many large packaged-software firms after 2000 do not cause further increases in the patent propensity trends depicted in Figure 5, suggesting a particularly significant drop in the number of patents issued to these firms after 2000.[30] These data for the largest packaged-software firms thus reinforce the

---

29 All patent propensity calculations use constant-dollar firm R&D spending (1987=$1).

30 Contractions are seen in R&D spending from 2000-2001 for several of the very large firms (Novell,

descriptive findings of Figure 2 that suggest that the post-2000 period is one in which patenting of software is if anything declining.

FIGURE 5 HERE

Inasmuch as electronic systems firms appear to account for a larger share of patenting during the 1987-2003 period than do packaged-software specialists, a comparison of patenting propensities between systems and software-specialist firms would be very interesting. Unfortunately, as we noted earlier, the absence of detailed line of business reporting of their R&D investments means that we have data on software-related R&D spending for only one of the 12 systems firms included in Figure 3, IBM. Figure 6 compares the patent propensities of IBM and Microsoft for the 1992-2002 period. The Figure is presented on a log scale, and shows that IBM's software patenting per software R&D dollar spent is substantially greater than Microsoft's, dominating Microsoft's propensity by a factor approaching or exceeding an order of magnitude (a factor of 10) in every 3-year interval. Furthermore, Microsoft's patent propensity has "plateaued" at 0.10-0.12 patents per $100 million during the 1996-2003 period, but IBM's has continued to grow, climbing from 0.7 patents per $100 million R&D during 1997-1999 to nearly 1.0 patents per $100 million R&D during 2001-2003.[31] Some of the reported growth in IBM's patent/R&D ratio reflects shrinkage in the firm's reported software R&D budget during 1997-2002, a period of growth for Microsoft R&D

Adobe Systems) as well among the smaller firms (Phoenix Technologies, Macromedia).

31 We include here two caveats given in Mowery and Graham (2003) concerning the risks of comparing these two firms' patent propensity: The R&D data reported by these two firms may not be strictly comparable, since a portion of Microsoft's total reported R&D investment may cover some fixed costs of maintaining an R&D facility that are not included in IBM's reported software-related R&D investment (Although IBM does maintain "software only" research facilities around the globe; Its Bangalore, India LINUX facilities are but one example). In addition, an unknown portion of Microsoft's reported R&D spending includes development programs for hardware—these data therefore may understate the Microsoft software-related patent propensity and overstate that for IBM.

investments. Nevertheless, the Figure suggests a considerable contrast in the patenting behavior of the largest packaged-software specialist and the largest software producer among U.S. electronic systems firms.

FIGURE 6 HERE

V. Patent quality issues

A. Evidence on trends in software patent "quality"

During the 1980s, some leading scholars argued that patents were preferable to copyright as a means of protecting software-related intellectual property, because of the higher "quality threshold" imposed by USPTO for the issue of patents, as well as the limited term of protection provided by patents. Menell's influential 1989 analysis of intellectual property protection of software, written in the wake of the strong judicial interpretation of copyright embodied in *Apple v. Franklin Computer Corp*., argued that patents had significant advantages over copyright as a means for protecting computer applications software, suggesting that "[t]he patent system's threshold requirements for protection—novelty, utility, and nonobviousness—are better tailored than the copyright standard to rewarding only those innovations that would not be forthcoming without protection" (Menell, 1989). The debate over the quality of software patents (see also Merges, 1999) centers on precisely these issues: Is the USPTO able to apply these requirements with sufficient rigor to prevent the issue of low-quality patents?

Defining, let alone measuring, the "quality" of patents issuing in a field of inventive activity is difficult. One measure of the technological or economic importance of a patent that has been widely employed in empirical work by economists and others (e.g., Trajtenberg, 1990) is the number of citations that a patent receives in subsequent patent applications after its issue.[32] The average number of citations received by a patent in the period shortly after its issue, therefore, may be interpreted as a measure of the importance of that patent. But patent-citation practices vary across fields and across time, making it difficult to compare the average number of citations received by software patents with those in such other fields as pharmaceuticals.

Our citation-based measure of patent importance addresses these issues by comparing the average number of citations received by software patents assigned to US firms, and various subpopulations within US firms, relative to the average number of citations received by all patents in our software classes (excluding the patents in the numerator of this ratio). This measure will not capture trends affecting all software patents, but it does enable us to depict trends in the average citations for patents assigned to (1) the 100 largest packaged-software specialist firms, as defined earlier; (2) our sample of 12 large electronic-systems firms' patents; and (3) all US-firm assignees for software patents, defined as above (this last class of software-patent assignees accounts for a steadily growing share of software patents during the 1987-2003 period, growing from 49% in 1987 to more than 62% in 2000). In order to bring our sample of patents as close to the present year as possible, we focus on average citations received within 3 years following issue, a relatively short "forward window." A value of less than unity for this ratio indicates that a given set of patents are cited less than average, while values greater than unity indicate relatively heavily cited patents.

The results of this descriptive exercise are depicted in Figure 7, covering citations within three years to patents issuing during 1987-2000, and do not suggest a precipitous

32 We exclude "self-citations" in this measure (citations from patents assigned to the assignee of the focal patent) in this analysis.

decline in the average number of citations received by patents in any of our three assignee classes. The measure of patent "quality" is lowest throughout the 1995-2003 period for patents assigned to our 12 electronic-systems firms, and highest for patents assigned to US firms overall. But there is no obvious trend of decline during 1987-2000 in the average citation ratio for software patents assigned to electronic systems firms, and the average citation ratio for patents assigned to US firms overall increases modestly during this period. The citation ratio for patents assigned to packaged-software specialists fluctuates widely, but also exhibits a modest upward trend. Although we do not have an explanation for these trends and differences among assignee classes, the lower quality of system-firms' patents (as well as these firms' larger share of software patenting overall) is consistent with the resort by these firms to "strategic patenting" (Hall and Ziedonis, 2001), a practice leading to their accumulation of large portfolios of patents of relatively low quality that can be used as "currency" in cross-licensing agreements to avoid costly litigation and/or injunctions.

FIGURE 7 HERE

B. Administrative remedies for software patent quality problems

Although this evidence for software does not suggest that the average quality of patents has declined sharply, the U.S. patent system faces challenges in reviewing applications in all new fields of inventive activity, largely because examiners rely heavily on patent-based prior art. This problem is by no means limited to software, although this field may present some novel twists, because the "patentability" of this particular artifact was endorsed by the U.S. judiciary well after the emergence of a substantial industry populated by inventive firms. The emergence of software patents thus occurred against a backdrop of substantial "prior art," much of which had not been patented.

A lack of efficient, reliable processes for resolving patent validity and ensuring higher patent quality also may slow the pace of invention in fields characterized by "cumulative invention," i.e., those in which one inventor's efforts rely on previous technical advances or advances in complementary technologies. But if these previous technical advances are covered by patents of dubious validity or excessive breadth, the costs to inventors of pursuing the inventions that rely on them may be so high as to discourage such "cumulative" invention. Alternatively, large numbers of low-quality patents may increase "fragmentation" of property rights covering prior generation or complementary technologies, raising the transaction costs for inventors of obtaining access (e.g., through licenses) to these technologies. Finally, the issue of a large number of low-quality patents will increase uncertainty among inventors concerning the level of protection enjoyed by these related inventions, which in turn will make it more costly and difficult for inventors to build on these related inventions in their own technical advances.

The issuance of low-quality patents also is likely to spur significant increases in patent applications, further straining the already overburdened examination processes of the USPTO. A kind of vicious circle may result, in which cursory examinations of patent applications result in the issue of low-quality patents, which in turn triggers rapid growth in applications, further taxing the limited resources of the USPTO, further limiting the examination of individual applications, and further degrading the quality of patents.

The U.S. patent system lacks strong administrative procedures for third-party challenges to patents before or immediately after their issue. The primary "quality control" mechanism for patent validity in the U.S. system is litigation, which is costly and time-consuming, often requiring years to resolve the validity of especially contentious (and therefore, valuable) patents. The USPTO "reexamination" procedure, originally envisioned as an alternative to expensive and time-consuming litigation, was created in 1980 as an administrative mechanism for administrative reviews of patent validity that would be less expensive and less time-consuming than litigation. A reexamination permits the patent owner or any other party to request that the USPTO reconsider the basis for issuing a patent, typically involving the disclosure to the Patent Office of some previously undisclosed "new" and relevant piece of prior art. Under the statute, a relevant disclosure must be printed in either a prior patent or prior publication—no other source can serve as grounds for the reexamination.

The reexamination procedure involves considerable costs and risks for parties seeking to invalidate an issued patent. The filing fee for the reexamination is substantial, and practitioners estimate the average costs of a reexamination at $10,000–$100,000 depending on the complexity of the matter. Although the costs of a reexamination are lower than those of litigation ($1–3 million), the third-party challenger in reexamination is denied a meaningful role in the process, and the patentholder maintains communications with the examining officer, offering amendments or adding new claims during the reexamination. Reexamination may make it more difficult for challengers to prevail in patent-validity litigation, because juries tend to give added weight to reexamined patents. Moreover, the Court of Appeals for the Federal Circuit has indicated that claims confirmed by the reexamining officer present added barriers to a successful court challenge to patent validity.[7] As a result, challengers face powerful incentives to forego reexamination in favor of litigation, a legal process that may well be more expensive, more time-consuming, and less expert in testing post-issue validity.

Members of the House of Representatives (U.S. Congress, 2001), the Federal Trade Commission (U.S. Federal Trade Commission, 2003), the USPTO (U.S. Patent and Trademark Office, 2003), the American Intellectual Property Law Association (American Intellectual Property Law Association, 2004), and the National Research Council (2004) all have expressed support for a post-grant review procedure to streamline the testing of patent validity, similar to the "opposition" procedure currently used by the European Patent Office. The opposition procedure of the European Patent Office differs from the U.S. reexamination process in almost all aspects, as Hall et al. (2004) point out. Reexaminations are much less common than oppositions, with an overall average patent challenge rate of 0.2%, in contrast to the European opposition rate of about 8%. Moreover, the party requesting a reexamination is the patent owner in at least 44% of the cases, lowering the effective rate even more. Although EPO oppositions may resolve validity challenges more slowly than USPTO reexamination proceedings in many cases (and almost certainly in important, complex cases with numerous opponents, appeals, etc.), the higher frequency of EPO opposition as compared to U.S. re-examination or litigation[33] suggests that the opposition process handles many more legal disputes over patent validity than are addressed by the U.S. reexamination process and at a lower cost than the U.S. litigation process.

VI. Conclusion

Spurred by favorable judicial decisions, software patenting has grown significantly in the United States since the 1980s, although the available data suggest that growth in software patents' share of overall U.S. patenting has slowed since

33 Allison, Lemley, Moore, and Trunkey (2003) have estimated that approximately 3% of issued patents were the subject of suits that were terminated during 1998-2000.

approximately 2000. Little evidence suggests that increased patenting has been associated with higher levels of innovation in the U.S. software industry, and equally little evidence suggests that increased patenting has proven harmful to innovation in this important sector of the "post-industrial" economy. The vertically specialized structure of the U.S. software industry, populated by firms specializing in software only, is a dramatic shift from the vertically integrated structure that characterized the U.S. and global computer industries in the 1960s. But stronger patent protection for software emerged in the 1980s, well after the transformation of this industry structure that began in the late 1960s. The links between stronger formal protection for intellectual property in this industry and the development of its vertically specialized structure thus are weak.

Patenting appears to have grown more rapidly than R&D spending during 1987-1999 in those firms for which both software patent data and software-specific R&D investment are available, although here too, growth has slowed since roughly 2000. But since the causal links among software-specific R&D investment, software innovation, and software patenting are murky or ambiguous at best, contentions that increased software patenting somehow are "causing" declines in R&D investment (see Bessen & Maskin, 2000; Bessen & Hunt, 2004) remain unproven. Indeed, the most robust conclusion seems to be that of Hall (2003):

> I[I]ntroducing or strengthening a patent system (lengthening the patent term, broadening subject matter coverage, etc.) unambiguously results in an increase in patenting and in the strategic use of patents. It is much less clear that these changes result in an increase in innovative activity, although they may redirect such activity toward things that are patentable and/or are not subject to being kept secret within the firm. (p. 10).

Electronic systems firms appear to account for a larger share of overall software patenting, in our definition, than do the packaged-software specialist firms during the 1987-2003 period. Moreover, the ratio of patents to software-specific R&D investment for the one large systems firm for which we have reliable data (IBM) suggests a significantly higher "patent propensity" than we observe for the largest packaged-software specialist firm, Microsoft. It is possible, although we have no direct evidence to support this argument, that systems firms are patenting their software-related intellectual property for strategic reasons, e.g., to support complex cross-licensing agreements similar to those in the semiconductor industry that are discussed in Hall and Ziedonis (2001). There is less evidence of such cross-licensing agreements among software specialists, although the recent agreement between Microsoft and Sun Microsystems (Guth and Clark, 2004) provides one such example. As Hall and Ziedonis note, much of the cross-licensing that provides incentives for extensive patenting by firms is motivated by the prospect or the reality of litigation. Evidence from software patent litigation cited in Graham (2004) indicates that packaged-software specialist firms account for a small fraction of software patent litigation, by comparison with computer hardware firms and firms from a diverse array of other industries.

One of the thorniest problems in analyzing software patenting, of course, is defining and measuring software patents. The empirical definition employed in this paper tends to weight packaged-software patents more heavily than definitions employed elsewhere (e.g., Bessen & Hunt, 2004). We employed this definition in order to track trends in patenting by software specialists, and it seems likely that the more expansive definition of "software patents" employed in Bessen & Hunt (2004) may categorize as software patents a number of patents in the fields of electronic controls that use "embedded" software in complex hardware systems. But the difficulties of defining and categorizing "software patents" point to a larger challenge to those who advocate prohibitions on software patents or some form of *sui generis* protection—if the definition of a software patent is open to debate and interpretation, how can rules (prohibitions or *sui generis* protection) that are specific to software inventions be defined and administered in a politically charged environment? Software is a genuinely "general purpose technology," and if anything, the boundaries defining what is and what is not a "software invention" are likely to become more blurry in the near future, further complicating efforts at developing regimes of intellectual property protection that are specific to software.

We have found little evidence that the "quality" of software patents has declined during at least the 1995-2003 period, although the evidence presented in this paper is limited in its scope and strength. The indicators of patent quality suggest that electronic systems firms are receiving lower-quality patents than packaged-software specialists, which could reflect greater resort to strategic patenting by the first group. Nonetheless, the problem of patent quality spans many fields of inventive activity in addition to software, as noted earlier, and is particularly acute for inventive activity in new technologies. Moreover, the U.S. patent system lacks the type of post-issue procedures (other than litigation) used by other industrial nations to strengthen patent quality. Rather than focusing on a "crisis" or other detrimental effects associated with software patenting *per se*, it seems more realistic and, in the long run, more beneficial to focus on developing better procedures to maintain high levels of patent quality in both new and established fields of inventive activity. The issue of software patent quality remains an important one, although evidence of sharp declines in the quality of recent patents in this field, assertions to the contrary notwithstanding, remains scarce. Nevertheless, the problem of patent quality in the U.S. system is by no means unique to software, but instead is common in many areas of new inventive activity. The relatively weak administrative procedures in the U.S. for ensuring patent quality (or for enabling administrative challenges by other interested private parties to patent validity) produce greater uncertainty over patent validity, and the only meaningful available mechanism (litigation) almost certainly results in higher costs overall for the resolution of such disputes. Rather than pursuing *sui generis* solutions to the challenges of patenting in software, a broader effort to strengthen administrative procedures for strengthening patent quality appears highly desirable.

VII. References

Abate, T., 1993. "Smaller, faster, better; Tech firms show off their latest wonders at trade show and foretell a user-friendly future," San Francisco Examiner, November 21, p. E:1.

Aharonian, G., 1993. "Setting the Record Straight on Patents," Communications of the ACM 36, pp. 17-18.

Allison, J., M. Lemley, K. Moore, and R. Trunkey, 2003. "Valuable Patents," mimeo, George Mason University.

American Intellectual Property Law Association, 2004. "AIPLA's Patent System for the 21st Century" (Washington, D.C.: AIPLA Report).

Association of University Technology Managers, *The AUTM Licensing Survey: Executive Summary and Selected Data, Fiscal Year 1999* (Norwalk, CT: Association of University Technology Managers, 2000: http://www.autm.net/surveys/99/survey99A.pdf).

Baba, Y., S. Takai, and Y. Mizuta, 1996. "The User-Driven Evolution of the Japanese Software Industry: The Case of Customized Software for Mainframes," in D.C. Mowery, ed., The International Computer Software Industry: A Comparative Study of Industry Evolution and Structure (New York: Oxford University Press).

Bessen, J., and E. Maskin, "Sequential Innovation, Patents, and Innovation," MIT Economics Department working paper, 2000.

Bessen, J., and R.M. Hunt, "An Empirical Look at Software Patents," unpublished MS, Boston University School of Law, 2004.

Bresnahan, T., and S. Greenstein, 1996. "The Competitive Crash in Large Scale Commercial Computing." in (eds) R. Landau, T. Taylor, and G. Wright, The Mosaic of Economic Growth, (Stanford, CA: Stanford University Press).

Campbell-Kelly, M., 1995. "Development and Structure of the International Software Industry, 1950-1990," Business and Economic History 24, pp. 73-110.

W.M. Cohen, R.R. Nelson and J.P. Walsh, 2001. "Protecting their Intellectual Assets: Appropriability Conditions and Why U.S. Manufacturing Firms Patent (or Not)," Working Paper 7552, National Bureau of Economic Research, Cambridge, Mass.

CONTU (National Commission on New Technological Uses of Copyrighted Works), 1979. Final Report. Washington: USGPO.

Graham, S.J., 2004. “Continuation, Complementarity, and Capturing Value: Three Studies Exploring Firms’ Complementary Uses of Appropriability Mechanisms in Technological Innovation,” unpublished Ph.D. dissertation, Haas School of Business, U.C. Berkeley.

Graham, S.J., and D.C. Mowery, 2003. “Intellectual Property Protection in the U.S. Software Industry,” in W. Cohen and S. Merrill, eds., The Patent System in the Knowledge-Based Economy (Washington, D.C.: National Academies Press).

Graham, S.J.H., and D.C. Mowery, 2004. “The Use of USPTO ‘Continuation’ Applications in the Patenting of Software: Implications for Free and Open Source.” *Forthcoming* Law and Society.

Graham, S.J.H., B.H. Hall, D. Harhoff, and D.C. Mowery, 2003. “Exploring the Effects of Patent Oppositions: A Comparative Study of U.S. and European Patents,” in W. Cohen and S. Merrill, eds., The Patent System in the Knowledge-Based Economy (Washington, D.C.: National Academies Press).

Guth, R.A., and D. Clark, 2004. “Behind Secret Settlement Talks: New Power of Tech Customers,” Wall Street Journal, April 5, p. 1.

Hall, B.H., 2003. “Business Method Patents and Innovation,” presented at the Atlanta Federal Reserve Bank conference on Business Method Patents, Sea Island, Georgia, April 3-5.

Hall, B.H., and R.M. Ziedonis, 2001. “The Patent Paradox in Semiconductors Revisited,” Rand Journal of Economics.

Hall, B.H., S.J.H. Graham, D Harhoff, and D.C. Mowery, 2004. “Prospects for Improving U.S. Patent Quality via Post-Grant Opposision,” in A. Jaffe, J. Lerner, and S. Stern, eds., Innovation Policy and the Economy, v. 4 (MIT Press for NBER).

Hart, R., P. Holmes, and J. Reid, 1999. “The Economic Impact of Patentability of Computer Programs: A Report to the European Commission,” Intellectual Property Institute, London.

Henderson, R., A.B. Jaffe, and M. Trajtenberg, 1998. "Universities as a Source of Commercial Technology: A Detailed Analysis of University Patenting, 1965-88," Review of Economics & Statistics, pp. 119-127.

Hunter, S.D., 2003. “Have Business Methods Patents Gotten a Bum Rap? Some Empirical Evidence,” MIT Sloan School working paper 4326-03, 2003.

Juliussen, E. & K. (eds.), 1996. The 8^{th} Annual Computer Industry Almanac (New York: Computer Industry Almanac, Inc.).

Kortum, S., and J. Lerner, 1999. "What is Behind the Recent Surge in Patenting?", Research Policy 28, pp. 1-28.

Kuan, J., 1999. "Understanding Open Source Software: A Nonprofit Competitive Threat," unpublished MS, Haas School of Business, UC Berkeley.

Lerner, J., and J. Tirole, 2000. "The Simple Economics of Open Source," unpublished MS, Harvard Business School.

Levin, R.C., A.K. Klevorick, R.R. Nelson, and S. Winter, 1987. "Appropriating the Returns to Industrial R&D," Brookings Papers on Economic Activity, 783-821.

Malerba, F., and S. Torrisi, 1996. "The Dynamics of Market Structure and Innovation in the Western European Software Industry," in D.C. Mowery, ed., The International Computer Software Industry: A Comparative Study of Industry Evolution and Structure(New York: Oxford University Press).

Maier, G., B. Lytle, and K. Ditthavong, 1999. "Patent protection provides long-term Net strategy," National Law Journal, October 18, p. B:11.

Markoff, J., 1993. "Patent Office to Review A Controversial Award," The New York Times, December 17, p. D:2.

Mansfield, E., 1985. "Patents and Innovation: An Empirical Study," Management Science 32,, 173-181.

Mazzoleni, R., and R.R. Nelson, 1998. "Economic Theories about the Benefits and Costs of Patents," Journal of Economic Issues 32, 1031-1052.

Menell, P., 1989. "An Analysis of the Scope of Copyright Protection for Application Programs." Stanford Law Review 41, pp. 1045-96.

Merges, R.P., 1996. "A Comparative Look at Intellectual Property Rights and the Software Industry," in D.C. Mowery, ed., The International Computer Software Industry: A Comparative Study of Industry Evolution and Structure (New York: Oxford University Press).

Merges, R.P., 1997. Patent Law and Policy (Charlottesville, VA: Michie).

Merges, R.P., 1999. "As Many as Six Impossible Patents Before Breakfast: Property Rights for Business Concepts and Patent System Reform." Berkeley Technology Law Journal 14, 578-615.

Mowery, D.C., R.R. Nelson, B. Sampat, and A.A. Ziedonis, 1999. "The Growth of Patenting and Licensing by U.S. Universities: An Assessment of the Effects of the Bayh-Dole Act of 1980," Research Policy, 2001.

Mowery, D.C., and T. Simcoe, 2002. "The Origins and Evolution of the Internet," in B. Steil, R. Nelson and D. Victor, eds., Technological Innovation and National Economic Performance (Princeton, NJ: Princeton University Press).

National Public Radio, 1998. "Patenting Math Formulas," All Things Considered, August 11, transcript 98081115-212.

National Research Council, Board on Science, Technology and Economic Policy, Committee on IPRs in the Knowledge-based Economy, 2004. "A Patent System for the 21st Century," S. Merrill, R. Levin, and M. Meyers, eds. (Washington, D.C.: NRC Report).

OECD, 1998. "Measuring Electronic Commerce: International Trade in Software," Directorate for Science, Technology, and Industry.

Orenstein, S., 1994. "U.S. Rejects Multimedia Patent," The Recorder, November 1, p. 4.

Peltz, J., 1993. "Compton's wins patent covering multimedia," Los Angeles Times, November 16, p. D:2.

Riordan, T., 1994. "Action Was Preliminary On a Disputed Patent," The New York Times, March 30, p. D:7.

Riordan, T., 1998. "Patents; An appeals court says a mathematical formula can be patented, if it is a moneymaker." The New York Times, August 3, p. D:2

Samuelson, P., 1990. "Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions." Emory Law Journal, 39, pp. 1025-1122.

Softletter, 1986. "The 1986 Softletter 100." Softletter Trends & Strategies in Software Publishing, 3, pp. 1-8.

Softletter, 1991. "The 1991 Softletter 100." Softletter Trends & Strategies in Software Publishing, 8, pp. 1-24.

Softletter, 1996. "The 1996 Softletter 100." Softletter Trends & Strategies in Software Publishing, 12, pp. 1-24.

Softletter, 1998. "The 1998 Softletter 100." Softletter Trends & Strategies in Software Publishing, 14, pp. 1-24.

Steinmueller, W.E., 1996. "The U.S. Software Industry: An Analysis and Interpretive History," in D.C. Mowery, ed., The International Computer Software Industry: A

Comparative Study of Industry Evolution and Structure (New York: Oxford University Press).

Trajtenberg, M., 1990. "A Penny for Your Quotes: Patent Citations and the Value of Innovations," Rand Journal of Economics 21, pp. 172-187.

U.S. Bureau of the Census, 1995. Service Annual Survey: 1995 (Washington, D.C.: USGPO).

U.S. Congress. House. Patent Improvement Act of 2001. H.R. 1333. 107th Cong., 1st sess. (April 3, 2001).

U.S. Department of Commerce, 1998. U.S. Industry and Trade Outlook 1998 (Washington, D.C.: USGPO).

U.S. Federal Trade Commission, 2003. "To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy" (Washington, D.C.: FTC Report).

U.S. International Trade Commission, 1995. A Competitive Assessment of the U.S. Computer Software Industry (Washington, D.C.: U.S. International Trade Commission).

U.S. Patent and Trademark Office, 2003. “21st Century Strategic Plan” (Washington, D.C.: USPTO Report).

U.S. Patent and Trademark Office, 2002. “Patenting by Organizations: 2001” (Washington, D.C.: USPTO Report).

Waldmeir, P. and L. Kehoe, 1999. "E-commerce companies sue to protect patents: Intellectual rights given legal test," Financial Times, October 25, p. 16.

Walsh, M., 1995. “Bowing to Reality, Software Maker Begins Building a Patent Portfolio,” The Recorder, Aug. 17, p. 1.

Figure 1: Software patent's share of all issued U.S. patents, 1987-2003

(Comparing two definitions: U.S. Classification and International Patent Classification)

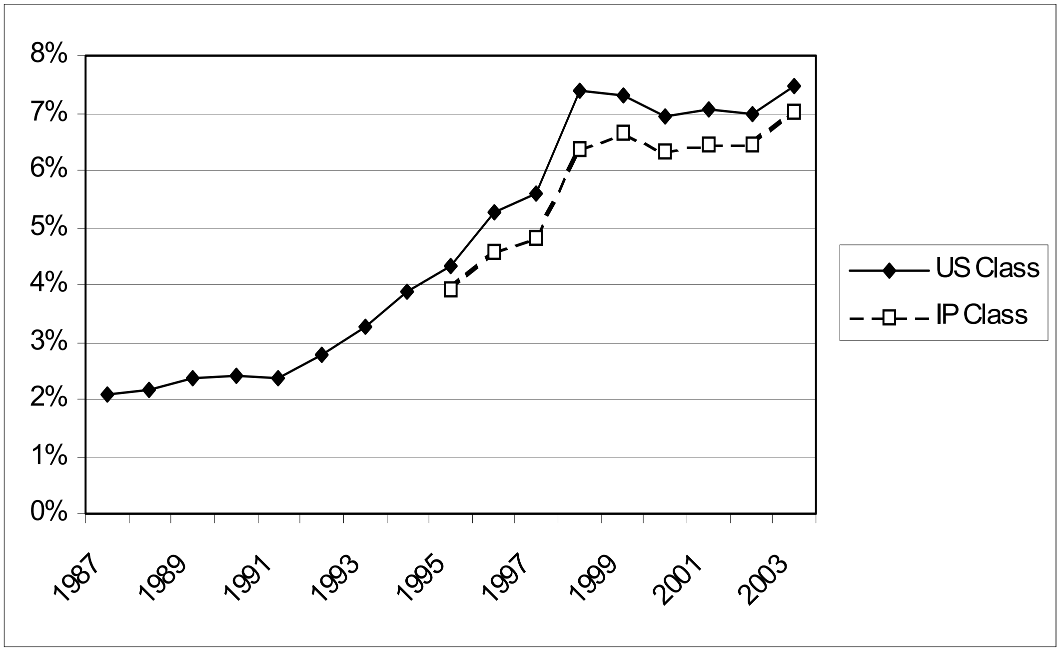

Figure 2: Large packaged-software firms' software patents, as a share of all US issued software patents, 1987-2003

(Comparison: US-class defined "software" patenting by 100 largest "packaged-software" firms as share of all "software" patents issued, *including and excluding* Microsoft)

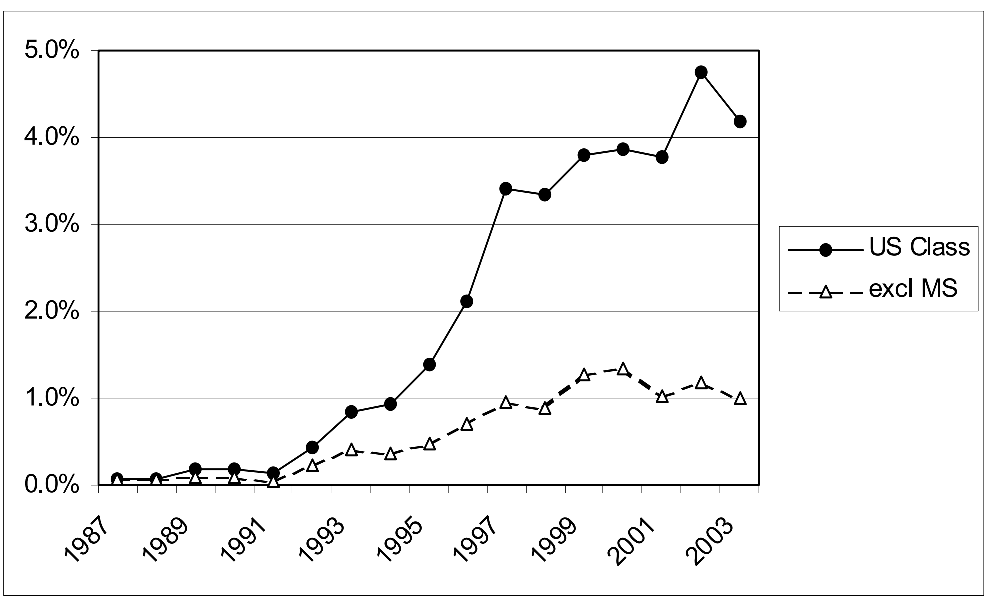

Figure 3: Large systems firms' software patents, as a share of all software patents, 1987-2003

(Weighted average: US-class defined "software" patenting by IBM, Intel, Hewlett-Packard, Motorolla, National Semiconductor, NEC, Digital Equipment, Compaq Computer, Hitachi, Fujitsu, Texas Instruments, and Toshiba)

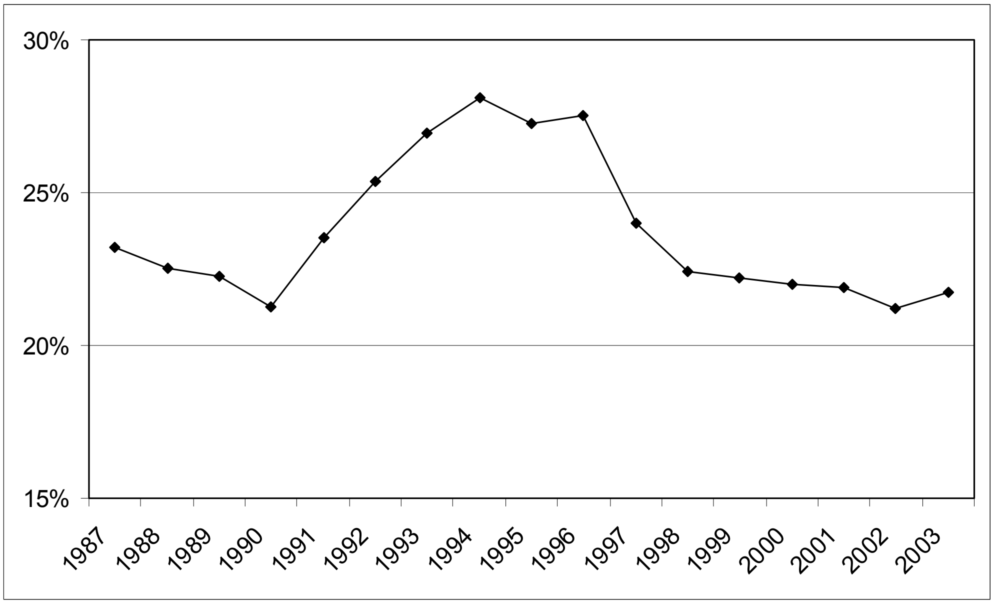

Figure 4: Large systems firms' software patents, as a share of firm patents, 1995-2003

(Weighted Average: Comparing US-class defined "software" patenting by IBM, Intel, Hewlett-Packard, Motorolla, National Semiconductor, NEC, Digital Equipment, Compaq Computer, Hitachi, Fujitsu, Texas Instruments, and Toshiba)

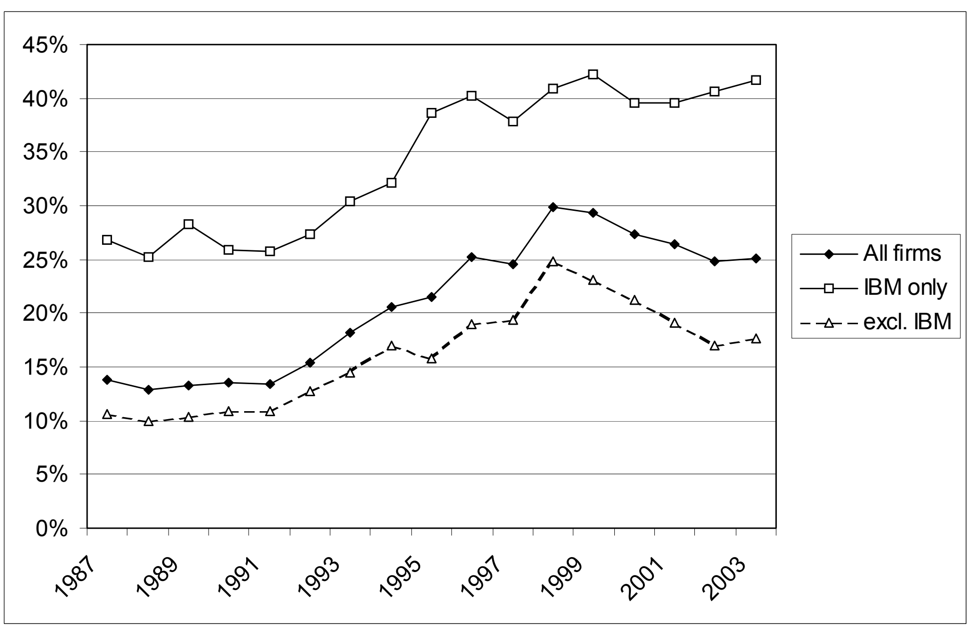

Figure 5: Top 15 packaged-software firms' software patent propensity, firms' patents per R&D expenditure, 1987-2002
(Comparing: Microsoft, Adobe Systems, Novell, Intuit, Autodesk, Symantec, Network Assoc., Citrix, Macromedia, Great Plains, RealNetworks, Borland (Inprise), SPSS, Phoenix Technologies, and Santa Cruz Operation with Top-14 firms [same, excluding Microsoft])

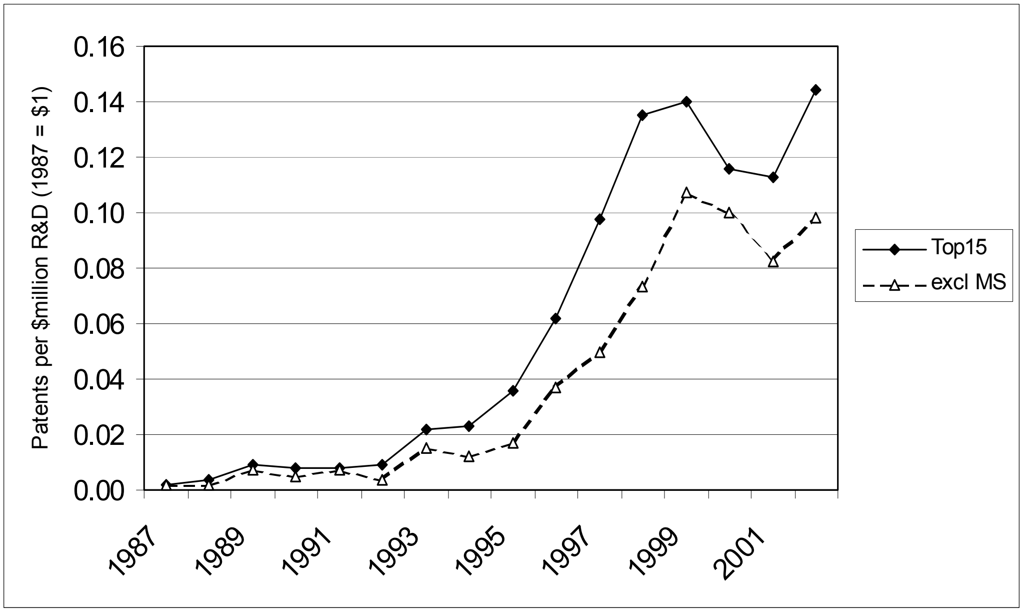

Figure 6: Comparison of IBM and Microsoft's software patent propensity, firms' "software" patents per "software" R&D expenditures, 1987-2002
(3-year moving averages; Patenting limited to each firms' defined software patents)

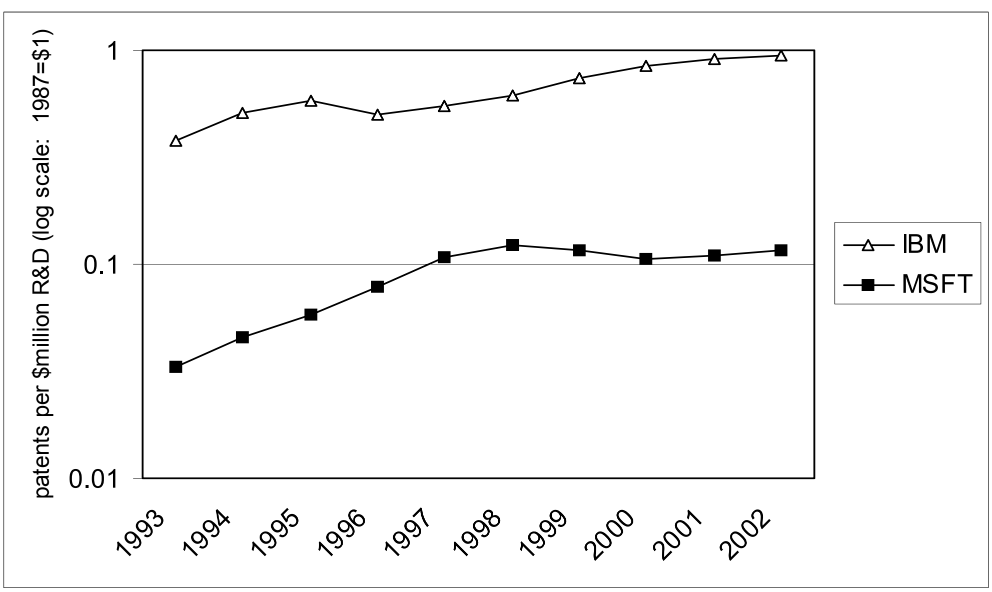

Figure 7: Forward citations ratio, by assignee class, 1995-2003:

Forward citations ratio for patents assigned to software specialist firms (SL100); electronic systems firms (Systems); and all US-firm assignees.