

# Software Patents, Incumbents, and Entry

## I. INTRODUCTION

Software patents have been controversial since the days when “software” referred to the crude programs that came free with an IBM mainframe. Different perspectives have been presented in judicial, legislative, and administrative fora over the years, and the press has paid as much attention to this issue as it has to any intellectual property topic during this time. All the while, a software industry began and has grown to a remarkable size, whether measured by revenues or profitability, number of firms or employees, or research expenditures. The scope of software innovation is even broader, as an increasing number of devices incorporate information technology, requiring modern manufacturing firms to employ programmers for increasingly diverse functions that often are performed more efficiently outside the software industry.

Although inventors have consistently asserted the need for patents to compete with industry incumbents, patent protection has not been easily and consistently available for much of this period. Rather, the legal system responded gradually to the advent of the industry, by broadening the scope and strength of protection for software-related inventions in fits and starts. The explosive growth of the industry is largely attributable to demand generated by the efficiency of software solutions; the expansion of the venture capital industry over the same period largely explains the lack of industry concentration.<sup>1</sup> The garage mentality also has a central explanatory force, as even some of the largest industry incumbents began with one or two (largely unfunded) inventors. And, there is

---

<sup>1</sup> The number of venture-capital investments in software firms increased rapidly during this period, from 11 in 1979, to 188 in 1989 to 1,035 in 1999. National Venture Capital Association, Yearbook (2004).

every reason to believe that increased patent protection has contributed to the ability of independent inventors and smaller firms to compete.<sup>2</sup>

Moreover, the ability to obtain patents on software always has been important to some of the industry incumbents, while others have exhibited little need for patents and, in some cases, strenuous opposition to the patentability of software. The incumbents are a diverse group. Some produce only software; others have substantial hardware product lines. Some sell to other technology firms and others sell applications to end users in a broad range of markets. And some sell prepackaged software products, while others focus on services – custom programming, installation, or maintenance. Regardless of the sector in which they participate, the incumbents spend massive amounts on research and development – about 14% of their annual revenues, more than \$60,000 per employee.<sup>3</sup> However, there are important patterns in patenting practices that the raw amount of investment in R&D cannot explain.

This Article examines the relation between patents and the different business models used by firms in the software industry. Our analysis has four parts. Part II provides a brief retrospective on software patents, emphasizing the shifting role of patents as the industry grew to its modern form. Part III uses quantitative data about patent portfolios to discuss the role that patents play for incumbent firms in the modern era. We highlight the fact that business models explain much of the pattern of patenting practices. Part IV describes the use of patents in the three channels through which

---

<sup>2</sup> The question of incentives is more difficult, given the important roles played both by young entrepreneurs who have earned millions and billions of dollars in this industry and by open-source developers driven, at least in part, by altruistic motives.

<sup>3</sup> See Corporate R&D Scorecard 2005, Tech. Rev., Sept. 2005, at 56.

technology flows into incumbent firms – venture-backed firms, open-source developers, and independent inventors, all of which contribute to the development of technologies that might supplant or improve the products and services currently delivered by incumbent firms. Finally, Part V concludes with a brief discussion of present-day industry perspectives on software patenting. As incumbents are now leading the charge on patent reform on all fronts,<sup>4</sup> we can expect that some change will occur. An understanding of the varying uses to which software firms put patents in their businesses provides a useful perspective on the types of reforms they advocate.

## **II. THE RISE OF SOFTWARE PATENTS**

### **A. Background**

Patent debates often focus on the statements and positions taken by noted industry participants.<sup>5</sup> Thus, it is common to hear that “even IBM once was opposed to patents,”

---

<sup>4</sup> E.g., Testimony of Mark Chandler, Senior Vice President and General Counsel, Cisco Systems, Inc., Subcommittee on Intellectual Property, Senate Committee on the Judiciary, Hearing on “Perspectives on Patents: Post-Grant Review Procedures and Litigation Reform” 1 (May 23, 2006) (testifying that “our patent litigation system is broken”); Brief of Business Software Alliance in Support of Petitioner, *eBay v. MercExchange, LLP*, 126 S. Ct. 1837 (2006) (No. 05-130); Brief of International Business Machines Corporation in Support of Neither Party, *eBay v. MercExchange, LLP*, 126 S. Ct. 1837 (2006) (No. 05-130); Brief of Business Software Alliance in Support of Petitioner, *KSR International v. Teleflex Inc.*, No. 04-1350 (cert. granted June 26, 2006); Brief of IBM Corporation in Support of Petitioner, *KSR International v. Teleflex Inc.*, No. 04-1350 (cert. granted June 26, 2006); The Peer to Patent Project: Community Patent Review (project sponsored by IBM at [www.communitypatent.org](http://www.communitypatent.org)).

<sup>5</sup> Surely the most discussed statements come from the PTO’s 1994 hearings on the propriety of software patents, where trenchant criticisms of software patents were made. Thus, the Principal Scientist at Adobe Systems, Incorporated testified unequivocally: “I believe that software per se should not be allowed patent protection. \* \* \* I argue that software should not be

or “Microsoft did fine without patents,” or even “we wouldn’t have Word, Excel, or PowerPoint if earlier inventors had been able to acquire patent protection.” There is a

---

patented, not because it is difficult to do so, but because it is wrong to do so.” Testimony of Douglas Brotz, Adobe Systems, Inc., United States Patent and Trademark Office, Public Hearing on Patent Protection for Software-Related Inventions 17 (Jan. 26, 1994) (San Jose, CA). A witness for Autodesk contended that “[t]here is absolutely no evidence whatsoever, not a single iota, that software patents have promoted or will promote progress.” Testimony of Jim Warren, Autodesk, Inc., United States Patent and Trademark Office, Public Hearing on Patent Protection for Software-Related Inventions 51 (Jan. 27, 1994) (San Jose, CA). Oracle’s Senior Vice President expressed a similar view. Testimony of Jerry Baker, Oracle Corporation, United States Patent and Trademark Office, Public Hearing on Patent Protection for Software-Related Inventions 25-26 (Jan. 26, 1994) (San Jose, CA).

On the other hand, major trade groups and software firms testifying at the same hearings expressed strongly held views that patent protection was important. So, for example, representatives of the Business Software Alliance (Testimony of Dianne Callan, Lotus Development Corporation, United States Patent and Trademark Office, Public Hearing on Patent Protection for Software-Related Inventions 18 (Feb. 10, 1994) (Arlington, VA), of the Computer and Business Equipment Manufacturing Association (Testimony of Richard LeFaivre, Apple Computer and Computer and Business Equipment Manufacturing Association, United States Patent and Trademark Office, Public Hearing on Patent Protection for Software-Related Inventions 8-9 (Jan. 26, 1994) (San Jose, CA)), and of Apple (Testimony of Richard LeFaivre, Apple Computer and Computer and Business Equipment Manufacturing Association, United States Patent and Trademark Office, Public Hearing on Patent Protection for Software-Related Inventions 8-9 (Jan. 26, 1994) (San Jose, CA)), IBM (Testimony of Victor Siber, Senior Corporate Counsel, IBM Corporation, United States Patent and Trademark Office, Public Hearing on Patent Protection for Software-Related Inventions 82-83 (Jan. 27, 1994) (San Jose, CA)), and Microsoft (Testimony of William Neukom, Microsoft Corporation, United States Patent and Trademark Office, Public Hearing on Patent Protection for Software-Related Inventions 70-71 (Jan. 27, 1994) (San Jose, CA)) all provided testimony supporting the patentability of software. Microsoft’s adherence to this position is notable in light of the \$120 million verdict entered against it a few days earlier in *Stac Electronics v. Microsoft Corp.*, No. 93-00413 (C.D. Cal. Jan. 23, 1994).

similar focus on market responses. Thus, prominent academics point to the limits on patent protection available to the software industry in its earliest days as evidence that software could succeed in the market without patent protection.<sup>6</sup> Of course, what we also see is that software patents did become important, largely because of market pressures. All of this raises the question whether the software industry in a “natural state” would be more competitive or innovative without patent protection.<sup>7</sup>

---

<sup>6</sup> E.g., Pamela Samuelson, *Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 *Emory L. J.* 1025, 1143 (1990) (“Thus, let us accept as a working assumption that the computer software industry has become a major industry without the aid of patents, and that had patents been in place in the industry's infancy, the field would not have grown as it has.”). Surveys of individual software engineers in the late 1980's suggest that this perspective was widespread among engineers at that time. See Samuelson & Glushko, *Comparing the Views of Lawyers and User Interface Designers on the Software Copyright “Look and Feel” Lawsuits*, 30 *Jurimetrics J.* 121, 140 (1989) (survey showing opposition to patent protection for various aspects of computer programs, including algorithms); see also Samuelson, *supra*, at 1148 (arguing that “if the software industry neither wants nor needs the patent system in order to be a vital and innovative industry, then, as a matter of public policy, it is sensible not to use the patent system for the protection of program-related innovations”).

<sup>7</sup> For example, in an article in *PC Magazine*, the columnist John Dvorak argued that software patenting is even bad for Microsoft (April 6, 2005), available at <http://www.pcmag.com/article2/0,1759,1781181,00.asp>. Still, recent scholarship provides at least some evidence that software patents have private value to the firms that obtain them. Iain M. Cockburn and Megan MacGarvie, *Entry, Exit And Patenting In The Software Industry* (NBER Working Paper No. 12563, 2006); Bronwyn H. Hall & Megan MacGarvie, *The Private Value of Software Patents* (NBER Working Paper No. 12195) (concluding that market evaluated software patents as unimportant ex ante, but ex post firms in the ICT sector that hold software patents are found to be valued at a significant premium relative to firms without software patents); Ronald J. Mann & Thomas W. Sager, *Patents, Venture Capital, and Software Startups*, *Research Pol'y* (forthcoming) (data indicating that patenting is related to progress of venture-backed software

Turning to the present, the narratives are more organized in some respects,<sup>8</sup> but less predictable in others.<sup>9</sup> There is also the question why technology firms have such divergent reactions to software patents, compared to say the pharmaceutical industry where there is a dominant consensus that vigorous patent enforcement is the best policy. Our discussion suggests a simple explanation for the different perspectives. Firms in the industry generally have supported software patents when it would be helpful to their

---

firms); Robert P. Merges, *Patents, Entry and Growth in the Software Industry*, available at <http://ssrn.com/abstract=926204> (debunking portfolio thesis by showing that patent effort by incumbent firms correlates closely with indicators of market success); Michael Noel & Mark Schankerman, *Strategic Patenting and Software Innovation*, available at <http://ssrn.com/abstract=922111> (empirical study of the effect of software patents on R&D and market value in the software industry, concluding that there are large positive technology spillovers from software R&D, but that patenting by rivals reduces R&D investment, patenting rates, and market value; finding a substantial patent premium in the market valuations of software firms).

<sup>8</sup> The most significant opponent of software patents in the United States is Richard Stallman and the Free Software Foundation; the Free Software Foundation often participates as an amicus in patent cases, and also presents testimony in congressional hearings. See “Opinion on Patent Retaliation,” available at <http://gplv3.fsf.org/patent-dd2.html>. That perspective is more widely held in the EU and elsewhere outside the United States, e.g., Florian Mueller, *Not Lobbyists as Such* (2005), where the opposition to software patents is bound up with balance-of-trade concerns. In the EU, for example, protection of software patents is generally regarded as transferring power and wealth from fledgling European firms to existing United States incumbent firms.

<sup>9</sup> As discussed below, incumbent firms have widely varying patenting strategies, which have led to different positions on important policy issues.

competitive position.<sup>10</sup> The divergent perspectives simply reflect divergent uses of patents for particular firms.

B. The Early Days: From Goetz to *Diehr*

In reality, software patenting predates the controversies of the 1990's, and indeed predates the software industry itself. Thus, although it might be a stretch to credit Samuel Morse with the first software patent,<sup>11</sup> it is plain that Bell Labs received an important software patent in 1951 for its "Error-Detecting and Correcting System."<sup>12</sup> Within the modern software industry, Martin Goetz's 1968 patent often is regarded as the first "true" software patent.<sup>13</sup> As Goetz's memoirs explain, the ability to patent his software was central to his firm's competitive position. Because Goetz was working at a period when IBM still bundled software with hardware – so that the software was effectively free if purchased from IBM – the only way of staking out a competitive foothold would be to prevent IBM from copying his product. A patent was the only apparent technique by which Goetz could obtain protection.<sup>14</sup> And those who have

---

<sup>10</sup> Despite high-profile cases requiring changes to Office, Windows, and Internet Explorer, Microsoft executives continue to extol the virtues of software patents. Bradford L. Smith and Susan O. Mann, *Innovation and Intellectual Property Protection in the Software Industry: An Emerging Role for Patents?*, 71 U. Chi. L. Rev. 241 (2004) (listing reasons why patent protection is superior to copyright).

<sup>11</sup> See Gregory A. Stobbs, *Software Patents* § 5.10[D][1], at 50-52 (2006 cumulative supplement).

<sup>12</sup> U.S. patent 2,552,629 (issued May 15, 1951).

<sup>13</sup> See *First Patent Is Issued for Software*, *Computerworld*, June 19, 1968, at 1 (reprinted in Martin Goetz, *Memoirs of a Software Pioneer: Part 1*, *IEEE Annals of the History of Computing*, Jan.-Mar. 2002, at 43, 51 (discussing U.S. patent 3,380,029, issued Apr. 28, 1968)).

<sup>14</sup> Goetz, *supra* note 13, at 50-53.

studied the market have concluded that the patent served its function well because its disclosure was so thorough as to give practical access to the technology to competitors that read it.<sup>15</sup>

IBM recognized the difficulty of obtaining IP protection for software, though its market position gave it a somewhat different perspective. It opposed unbundling its software because of “our present inability to protect the proprietary use of our programming systems. \* \* \* We must settle on whether or not, and to what degree, we can protect programs before we can deal adequately with the question of selling them.”<sup>16</sup> As long as its software was bundled, IBM regularly took the position that patent protection for software was inappropriate.<sup>17</sup> The parallel to IBM’s modern involvement with Linux is startling – a market strategy in which IBM would profit from the sales of proprietary hardware in a value chain joined with free software.<sup>18</sup>

Unfortunately for IBM’s competitors, the industry’s efforts to clarify the scope of patent protection during the 1970’s were generally not fruitful. As the decade began, the case for software patenting was a strong one, based on the 1969 decision of the Court of

---

<sup>15</sup> Martin Campbell-Kelly, *Not All Bad: An Historical Perspective on Software Patents*, 11 Mich. Telecomm. Tech. L. Rev. 191, 214-15 (2005).

<sup>16</sup> See Stobbs, *supra* note 11, § 1.04[A], at 39 (quoting testimony from 1960’s IBM antitrust litigation).

<sup>17</sup> See U.S. Patent Court to Rehear Software Issue, N.Y. Times, Mar. 1, 1969, at 43 (discussing IBM’s opposition to software patents in proceedings before the PTO, aligned against Goetz’s employer Applied Data Research); see also Samuelson, *supra* note 6, at 1143 (noting early opposition to software patents by IBM and other leading hardware firms).

<sup>18</sup> See Ronald J. Mann, *Commercializing Open Source Software: Do Property Rights Still Matter?*, 20 Harv. J.L. & Tech. (forthcoming 2006).



Customs and Patent Appeals in *In re Prater*.<sup>19</sup> The Supreme Court's 1972 decision in *Gottschalk v. Benson*,<sup>20</sup> however, brought *Prater* into grave doubt by invalidating a patent on an algorithm for converting binary-coded decimal numerals into pure binary numbers. In the Supreme Court, IBM and other large hardware manufacturers opposed to the patent squared off against trade groups representing small software firms trying to gain a foothold in the industry.<sup>21</sup> Because the patent in question was regarded as weak by firms in the industry, the victory was not regarded as catastrophic, although it did diminish the enthusiasm for patent filings at the time.<sup>22</sup> The 1978 decision in *Parker v. Flook*,<sup>23</sup> invalidating a method for updating an alarm limit in a chemical process, did not make things any better, though again the narrowness of the decision left the industry uncertain of the ultimate question of patentability.<sup>24</sup> Still, many observers at the time

---

<sup>19</sup> 56 C.C.P.A. 1381, 1394-98 (1969).

<sup>20</sup> 409 U.S. 63 (1972).

<sup>21</sup> See Warren Weaver Jr., High Court Denies Computer Patent for Programing [sic], N.Y. Times, Nov. 21, 1972, at 1.

<sup>22</sup> See Martin Goetz, Memoirs of a Software Pioneer: Part 1, IEEE Annals of the History of Computing, Oct.-Dec. 2002, at 14, 18 (suggesting that the patent was "not representative" of inventions in the industry and "never should have been filed or appealed"). For a close reading of *Benson*, arguing that it is much more hostile to software patenting than Goetz suggests, see Samuelson, *supra* note 6, at 1048-62.

<sup>23</sup> 437 U.S. 584 (1978).

<sup>24</sup> See Goetz, *supra* note 22, at 22 (noting the perceived ambiguity of *Parker*); N.R. Kleinfeld, Software Patent Issue Is Murky, N.Y. Times, June 28, 1978 (reporting Goetz's view that "[n]one of the computer programs that came before the Supreme Court is regarded by the software industry as a good example of high-level programming").

regarded *Parker* as a major setback.<sup>25</sup>

Thus, through the 1970's and into the 1980's, many firms routinely failed to patent inventions that readily would have been patented in later decades. Martin Campbell-Kelly argues forcefully that the lack of clear patent protection had adverse effects on the pace of development. Offering an example regarding the invention of VisiCalc (a prominent early spreadsheet), he argues that the spreadsheet sector would have developed more rapidly if the inventor had patented the technology, because competitors would have had access to the patent disclosure as a way to understand the technology instead of "reinventing the wheel" for themselves.<sup>26</sup> Similarly, he contends that the proliferation of word processors by the early 1980's reflected wasteful "overfishing" that would have been truncated if early innovators had obtained effective patent protection.<sup>27</sup>

By the end of the 1970's IBM's position had changed. After it unbundled its software from its hardware in 1970, IBM entered the field of competition, in which it had to sell its software products, at a separate price, and attempt to defend them from appropriation by competitors. Pressure from newcomers to the industry, particularly

---

<sup>25</sup> See Linda Greenhouse, *Court Curbs Software Patents*, N.Y. Times, June 23, 1978, at D1 (quoting software trade association representative who argued that "the [software] industry \* \* \* now has no Federal protection against theft by competitors," which he viewed as a "gigantic industry problem"); see also Samuelson, *supra* note 6, at 1076-83 (detailed contextual analysis of *Flook*).

<sup>26</sup> Campbell-Kelly, *supra* note 15, at 198-99.

<sup>27</sup> Campbell-Kelly, *supra* note 15, at 201-02, 209.

those from Japan,<sup>28</sup> quickly turned IBM into an advocate for increased IP protection for software.<sup>29</sup> And with the Court's 1981 decision in *Diamond v. Diehr*,<sup>30</sup> the way seemed open though perhaps not yet clear, for regular patenting of the kinds of computer programs that were at the core of the rapidly growing software industry.<sup>31</sup>

C. The 1980's: The PC and the First Software Patent Portfolios

The Court's 1981 decision in *Diehr* certainly was not the most important event of the year for the software industry. IBM's introduction of the personal computer was much more significant. Although others had sold personal computers with some success – Apple and Tandy at the time were regarded as formidable competitors – the introduction of the IBM PC transformed the software industry.<sup>32</sup> For one thing, the deployment of the IBM PC and the rapid entry of parallel IBM-compatible machines<sup>33</sup>

---

<sup>28</sup> Japanese software developers benefited from the same government support as competitors in other industries on which Japan focused. See Steve Lohr, Japan's Hard Look at Software, N.Y. Times, Jan. 9, 1983.

<sup>29</sup> See Goetz, *supra* note 22, at 24-28; Angel Castillo, Bill Safeguards Data Programs, N.Y. Times, Dec. 4, 1980, at D1 (noting IBM's support for the Computer Software Copyright Act of 1980). IBM's first major IP claim in the software industry came at this time. Tellingly, IBM challenged Hitachi's incorporation into its software products of technology it claimed Hitachi had stolen from IBM. Hitachi Disputes Fee to I.B.M., N.Y. Times, Nov. 10, 1983.

<sup>30</sup> 450 U.S. 175 (1981).

<sup>31</sup> See Goetz, *supra* note 22, at 22-23 (noting positive contemporaneous response to *Diehr*).

<sup>32</sup> See Martin Campbell-Kelly & William Aspray, *Computer* (1997); Andrew Pollack, Big I.B.M.'s Little Computer [hereinafter Pollack, Big IBM], N.Y. Times, Aug. 13, 1981; Andrew Pollack, Next, a Computer on Every Desk, N.Y. Times, Aug. 23, 1981.

<sup>33</sup> See Andrew Pollack, Big I.B.M. Has Done It Again, N.Y. Times, Mar. 27, 1983 (discussing the proliferation of IBM clones).

fostered a competition for a standard operating system to be used on those machines, a competition in which Microsoft's MS-DOS system successfully dislodged the CP/M system developed by then market leader Digital Research, one of many companies fated to become bywords for a lack of market foresight in this rapidly developing industry.<sup>34</sup> As Figures One and Two display, the need to protect that operating system would make Microsoft one of the first software products firms to invest heavily in patents as a way to protect its core technology.<sup>35</sup>

The other thing that came from the deployment of the PC was a vast and previously unimaginable market for software applications to be deployed on the geometrically increasing number of personal computers deployed in the American workplace.<sup>36</sup> This signaled the end (or at least diminished importance) of the "garage" era of software development, as the need to produce sophisticated applications rapidly called into existence a large number of large firms, all of whom began to compete against each other for the attention of the limited available capital investors.<sup>37</sup> Not surprisingly,

---

<sup>34</sup> A common perspective is that CP/M failed because Digital Research moved too slowly to upgrade its software to accommodate 16-bit processors, leaving Microsoft's MS-DOS to gain an insuperable lead in that market before Digital Research ever entered. See Martin Campbell-Kelly, *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry* (2003); David E. Sanger, *The Big Guys Get into the Act*, N.Y. Times, Sept. 4, 1983.

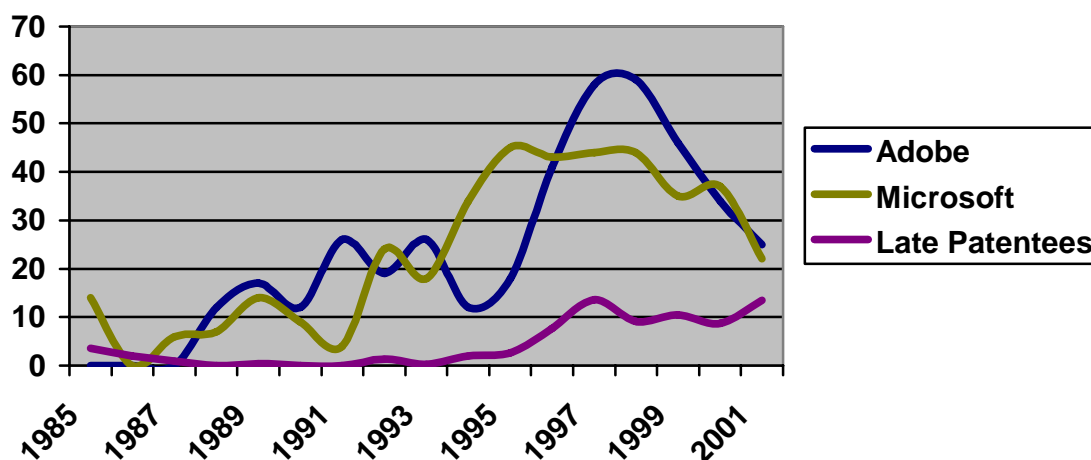
<sup>35</sup> See Stobbs, *supra* note 11, § 11.01[B] (discussing early Microsoft patents on technology related to operating systems).

<sup>36</sup> See Campbell-Kelly, *supra* note 34; Pollack, *Big IBM*, *supra* note 32 (discussing IBM's plans to foster widespread development of software for the IBM PC).

<sup>37</sup> See Andrew Pollack, *Slugging It out on the Software Front*, N.Y. Times, Oct. 16, 1983; Andrew Pollack, *Microsoft Has It All-Almost*, N.Y. Times, Sept. 4, 1985 (discussing market forces leading to "professionalization" of software development).

the fiercely competitive landscape of rapid development against a backdrop of uncertainty in legal rules led firms to adopt very different IP strategies. Adobe, like Microsoft, began to patent relatively early, apparently hoping to protect its early lead in the so-called “Font Wars” of the late 1980’s.<sup>38</sup>

**Figure One: Patent Applications<sup>39</sup>/B\$ Sales (Prepackaged Software) (1985-2001)**

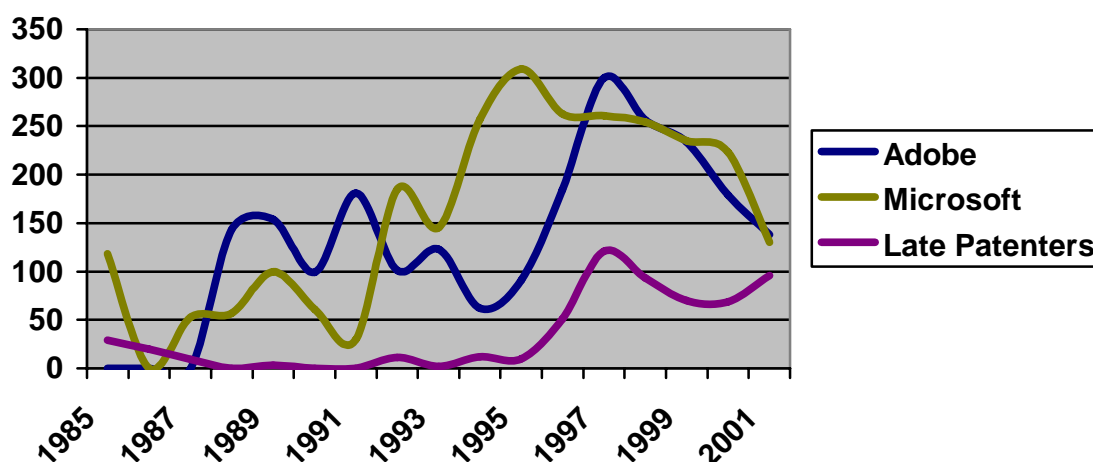


---

<sup>38</sup> See Andrew Pollack, Adobe Is Set to Disclose Technology, N.Y. Times, Sept. 21, 1989; Peter H. Lewis, The Fallout from the Font Wars, N.Y. Times, Oct. 1, 1989.

<sup>39</sup> Throughout this paper, the tables and figures attribute applications to the year in which they were filed and refer only to patent applications that resulted in issued patents.

**Figure Two: Patent Applications/B\$ R&D (Prepackaged Software) (1985-2001)**



For the most part, however, products firms during the 1980's eschewed patent protection, apparently accepting the predictions of the CONTU-like pundits advising that copyright law would provide adequate protection. Thus, Figures One and Two group together under the title of "Late Patentees" Autodesk, Computer Associates, and Oracle (the three leading products firms of the time that have survived and have substantial patent portfolios today). Two obvious factors can explain the change of strategy. The first would be the belated<sup>40</sup> discovery (as the patent applications filed in the 1980's matured into issued patents) that some of their competitors already were amassing substantial portfolios.<sup>41</sup> There was considerable uncertainty about the reliability of

<sup>40</sup> One reason that the discovery was "belated" is that the PTO seems to have dragged its feet considerably in response to the patent applications that the software firms started filing in the mid-1980's, which had the effect of increasing the head-start of applications that those firms had by the time their competitors learned of the applications.

<sup>41</sup> The tone of surprise is evident from the press coverage that greeted a 1989 patent by Quarterdeck on an early form of "windows" technology. See Patent Is Won by Quarterdeck,

software patents even after *Diehr*,<sup>42</sup> and firms that saw them as less crucial than Microsoft and Adobe easily could (and did) forego them. The other is the decision in *Lotus v. Borland*<sup>43</sup> signaling the limited ongoing reliability of copyright as a system for protecting innovation in software.<sup>44</sup>

#### D. The 1990's: Proliferation of Software Patents

As others already have documented in detail, the 1990's brought a rapid acceleration in the growth of patents in the software industry, as the problems with copyright became more evident, as the legal environment became more supportive, and as the pace of software innovation grew with the spread of the personal computer and

---

N.Y. Times, Apr. 19, 1989 (suggesting, wrongly as it turns out, that it is “unusual” to receive a software patent and that the patent could be more important than Apple’s battle with Microsoft over copyright protection for its graphic user interface)

<sup>42</sup> For example, Donald Chisum argued vehemently that *Benson* needed to be explicitly overruled to clear the way for a sensible system of software patenting. Donald Chisum, The Patentability of Algorithms, 47 U. Pitt. L. Rev. 959 (1986). On the other side of the issue, Pam Samuelson argued in 1990 that the PTO had gone too far to tolerate software patents and that the courts needed to step in to prevent the proliferation of patents in the area. Samuelson, *supra* note 6.

<sup>43</sup> *Lotus Development Corp. v. Borland Int’l, Inc.*, 49 F.3d 807 (1<sup>st</sup> Cir. 1995), *aff’d* by an equally divided Court, 516 U.S. 233 (1996).

<sup>44</sup> See Lerner, Joshua and Zhu, Feng, “What is the Impact of Software Patent Shifts?: Evidence from *Lotus v. Borland*” (March 2005). NBER Working Paper No. W11168 (examining impact of *Lotus* and presenting evidence that reduction in copyright protection led to more patenting by interface firms).

then the Internet. This produced a rate of patenting that was unimaginable to those in the industry just a few decades earlier.<sup>45</sup>

To see how those factors changed the relative patenting rates among various types of software firms, we present in Figures Three and Four data on the patenting practices from 1990-2001 of three groups of large firms that survived throughout the entire period and now have substantial software patenting portfolios: electronics firms, prepackaged software firms, and system design firms.<sup>46</sup> As those Figures illustrate (with alternate data on patents as a function of sales and as a function of R&D), electronics firms already had established stable patenting practices and thus experienced only a modest rise in patenting rates from 1990-2001, while the firms in the other two sectors, more focused on software, experienced an increase of 300-500%. The point is illustrated more clearly in Figures Five, which shows the patenting rates for one prominent firm from each sector: HP's line has only a slight upward trend, while the Microsoft and IBM lines show much steeper increases that are surprisingly parallel given the common perception that Microsoft is a relative latecomer to the patent sweepstakes.

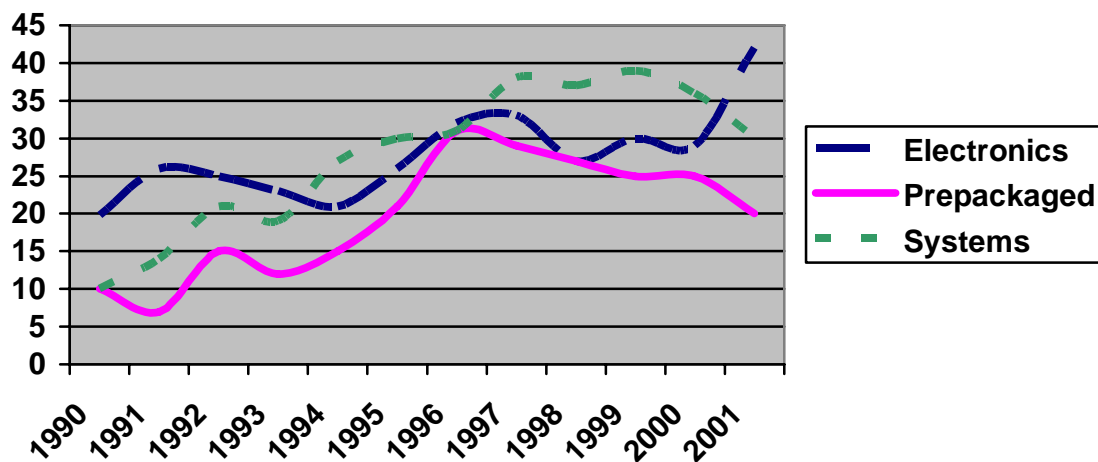
---

<sup>45</sup> See Kleinfeld, *supra* note 24 (quoting Goetz's comment: "If the Patent Office were to become receptive to giving out patents on software, I doubt that there would be more than a few hundred applications a year.").

<sup>46</sup> The firms are distinguished by three-digit NAICS codes: 334 for the electronics firms (Apple, EMC, HP, NCR, Qualcomm, and Sun), 511 for the prepackaged software firms (Adobe, Autodesk, Computer Associates, Microsoft, Oracle, Sybase, and Synopsys), and 541 for system design and processing (EDS, IBM, Mentor Graphics, Novell, and Unisys).



**Figure Three: Patent Applications/\$B Sales (by Sector)**



**Figure Four: Patent Applications/\$B R&D (by Sector)**

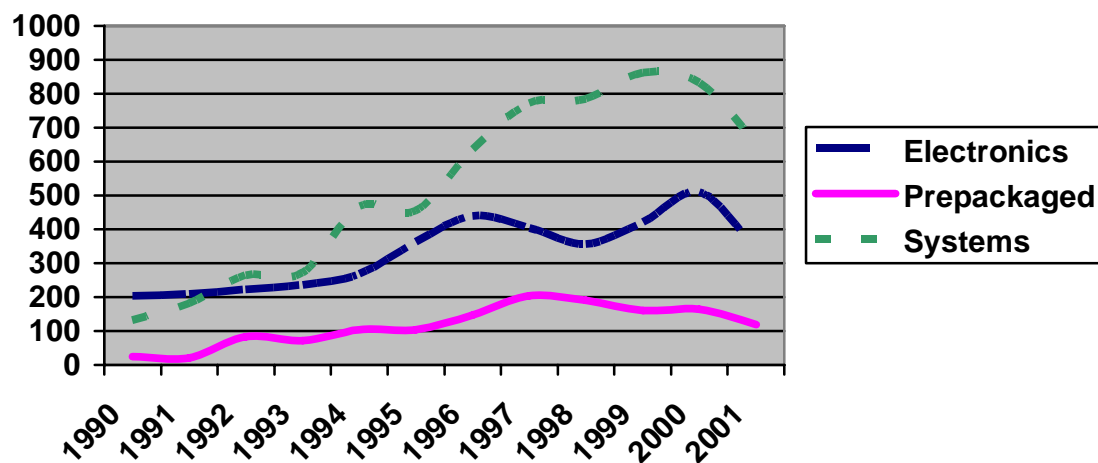


Figure Five: Patent Applications/\$B Sales (by Firm)

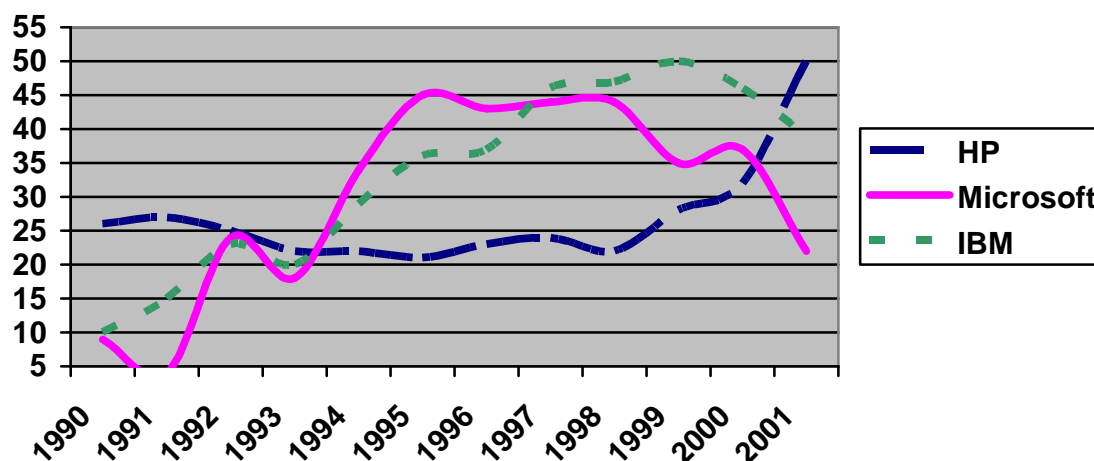
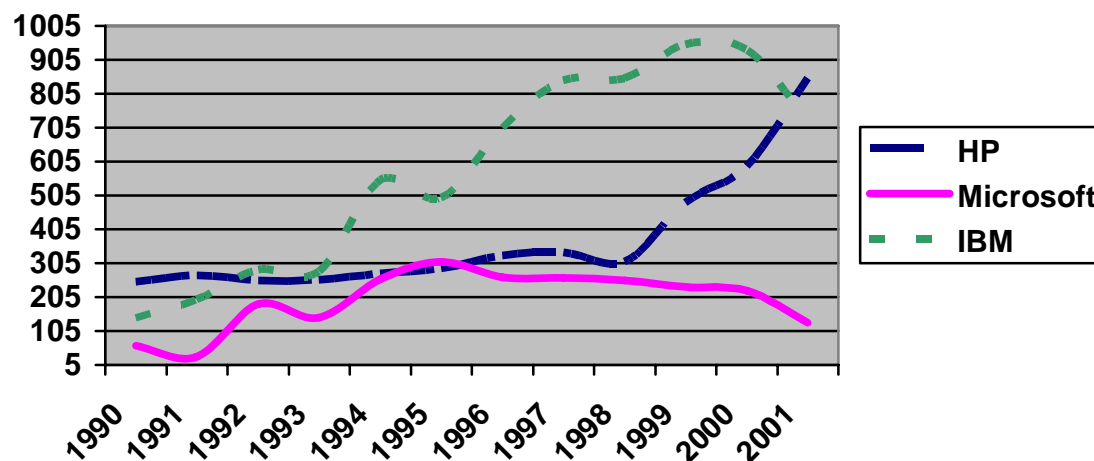


Figure Six: Patent Applications/\$B R&D (by Firm)



The data on patenting as a function of R&D is particularly illuminating, given the industry focus on the ratio of patenting to R&D dollars.<sup>47</sup> Indeed, some academics have become concerned that patents might be substituting for research expenditures.<sup>48</sup> Others,

<sup>47</sup> Randall Stross, Why Bill Gates Wants 3,000 New Patents, NYT 7/31/05.

<sup>48</sup> Bessen and Hunt, supra note 55 (unpublished manuscript arguing that software patents substitute for R&D spending); see also Robert M. Hunt, When Do More Patents Reduce R&D?, 96 Am. Econ. Rev. 87 (2006).

however, have pointed out that patenting and R&D affect one another and are driven by similar factors.<sup>49</sup>

What we do know (as illustrated by the descriptive data in Figures Two, Four, and Six above) is that the propensity to patent in the industry increased rapidly during the early part of the 1990's. It is evident, however, that the rapid upward trend ended quite some time ago, perhaps by 1997. Indeed, if the comparative charts suggest anything, it is that R&D expenditures as a function of sales have grown more rapidly in the prepackaged software sector than in the others.

In general, however, none of this tells us whether firms are innovating more or less, or whether the patents are of a higher or lower quality. Rather, these charts suggest that the firms in the industry, for the reasons discussed above, were steadily instituting processes to protect more of their technologies and protecting things that might be less “valuable” or less central to their existing products. Thus, the shift in propensity to patent might reflect a conscious decision to protect more of the things that the firm has developed that are not yet marketed (and thus protected through secrecy). In short, where firms in the 1980's patented their most fundamental and crucial technologies, the modern software firm with a patenting portfolio is likely to patent as a matter of routine.<sup>50</sup>

### **III. EXPLAINING THE PATTERN OF PATENTS IN THE MODERN SOFTWARE INDUSTRY**

#### **A. Introduction**

---

<sup>49</sup> Ashish Arora, Marco Ceccagnoli, and Wesley M. Cohen, R&D and Patent Premium (NBER Working Paper No. 9431, 2003).

<sup>50</sup> Along the way, the industry has transitioned from one with a small number of relatively broad “pioneering patents,” to one with a large number of narrow patents, which has caused justifiable concern about the transaction costs of licensing.

Turning from history to the modern software industry, what we see now is a complex pattern of software patenting influenced by features of the firm. Although the frequency of patenting is a major topic of public debate, little work has been done to explain the pattern of patenting.<sup>51</sup> To be sure, the existing literature does recognize a rough cross-licensing equilibrium among the incumbent firms.<sup>52</sup> Many of those firms have substantial patent portfolios, but a web of explicit or implicit cross-licensing agreements means that the major incumbents have access to most of the patented technologies in the industry. Thus, those firms compete against each other for the most part on product design and marketing. Patent-based market power is not a significant factor.

If the desire to build portfolios for defensive purposes were the main justification for patents in the industry, however, you would expect portfolios roughly proportionate to litigation exposure. Assuming that the firm's size is a reasonable proxy for litigation exposure, this suggests an easy quantitative inquiry. Specifically, if the defensive

---

<sup>51</sup> Although the literature provides no definitive theoretical framework for predicting when patents will be useful, a paper by Wesley Cohen and his co-authors takes steps toward a general explanation as part of a description of differences between the United States and Japan. Wesley M. Cohen et al., *R&D Spillovers, Patents, and the Incentives to Innovate in Japan and the United States*, 31 *Research Pol'y* 1349 (2002). In their view, patents can play two distinct roles: as tools for exclusion (to be exploited through production within the patenting firm), and as tools for licensing (to be exploited through licensing outside the boundary of the patenting firm). They develop a distinction between "discrete" and "complex" products, finding evidence to support the idea that "complex" product industries in the United States rely more heavily on licensing to permit exploitation outside the boundaries of the firm.

<sup>52</sup> For detailed discussion, see Ronald J. Mann, *Do Patents Facilitate Financing in the Software Industry?*, 83 *Tex. L. Rev.* 961 (2005).

portfolio hypothesis is correct, patent portfolios would correlate closely with size and there would not be a great deal of variation tied to other factors such as market sector or R&D intensity.

Because there are in fact notable differences in patenting practices in different sectors of the software industry,<sup>53</sup> we expect that the pattern of patenting will depend not only on size, but also on whether the firm focuses on selling products or services, how devoted the firm is to R&D (conventionally measured by R&D intensity, calculated as \$R&D/employee), whether the firm is primarily a software firm or a hardware/electronics firm, and competitive issues in the specific sector of the software industry in which the firm is located.

To examine those questions, we combined patent data with data about firms in *Software Magazine's* Software 500 from 1998-2002.<sup>54</sup> Because we are interested in the

---

<sup>53</sup> See Mann & Sager, *supra* note 7.

<sup>54</sup> The Software 500 purports to include the top 500 firms in the software industry each year by software revenue. Software Magazine collects information for the Software 500 from an annual vendor survey, public documents, press releases, SEC filings, and industry analysts. Based on interviews within the industry, we have the impression that the survey response rate is quite high. The list appears to be widely regarded as authoritative within the industry. Campbell-Kelly, for example, uses the list pervasively in his comprehensive history of the industry. Campbell-Kelly, *supra* note 34. Among other things, it is considerably more comprehensive than the Softletter 100, which is limited to prepackaged software providers and thus generally excludes services firms. See Stewart J.H. Graham & David C. Mowery, Intellectual Property Protection in the U.S. Software Industry, in *Patents in the Knowledge-Based Economy* 219 (Wesley M. Cohen & Stephen A. Merrill eds., 2003).

Because of considerable turnover in the industry, that list includes about 1000 firms for the five-year period. For each firm, the Software 500 includes several data points of interest, total revenues, total revenues from software-related activities, % of revenues expended on research and

characteristics that relate to software patents, we divided the dataset of 34,000 patents into software and non-software patents. The methodology was to examine the patents individually to determine whether each patent, properly speaking, should be considered a patent on a software invention.<sup>55</sup> Using that methodology, about 68% (13,500) of the non-IBM patents qualified as software patents and about 55% of the IBM patents (extrapolating from the sample that we examined), for a blended total of about 62% (21,200) software patents.<sup>56</sup> As Table One illustrates, the patents were highly concentrated – more than 80% of the firms had not even a single patent, and less than 10% had more than one patent.

---

development, number of employees, and % of revenues generated by the sale of services. Because the purpose of our study is to focus on firms that fairly can be characterized as software firms, we excluded the 18 firms that did not derive at least 20% of their total revenues from software in any of the five years for which we collected data.

<sup>55</sup> Allison examined all of the patents except the patents for IBM and categorized as a software patent each patent for which at least one claim element involved data processing. For the 14,000 IBM patents, he read a random sample of about 300 patents and extrapolated from that sample. The distinction is a difficult one because there is no specific patent class for software patents. Prior scholars have taken one of two approaches. First, Graham and Mowery look at the portfolios of large prepackaged software firms and develop a set of classes that includes most of their patents. Graham & Mowery, *supra* note 54. Second, Bessen & Hunt develop a keyword search designed to capture software patents. James Bessen and Robert M. Hunt, *An Empirical Look at Software Patents (Research on Innovation)*, available from <http://www.researchoninnovation.org/swpat.pdf>. Although our approach arguably is more subjective, we believe that the increased accuracy makes it preferable.

<sup>56</sup> To provide additional data points for robustness checks (as described below), we subsequently collected a set of all of the patents issued to the firms from January 1, 2003 through June 30, 2005 (an additional 20,000 patents), but we did not analyze those patents to determine whether they were software patents or not.

**Table One: Distribution of Patent Portfolios**

	Software Patents	All Patents
	(1998 and 1999)	(1998-2000)
# of Patents	# of Obs. (Cum. %)	# of Obs. (Cum. %)
0	596 (85%)	899 (81%)
1	49 (92%)	90 (89%)
2	14 (94%)	30 (91%)
3	5 (94%)	17 (93%)
4	6 (95%)	16 (94%)
5	0 (95%)	3 (95%)
6	1 (95%)	2 (95%)
7	4 (96%)	2 (95%)
8	0 (96%)	3 (95%)
9	1 (96%)	2 (96%)
10 or more	27 (100%)	50 (100%)
Total # Obs.	703	1114

We examined the variation in patenting by constructing a patent production function to identify the factors that might influence the number of patents a firm produces.<sup>57</sup> Because we are interested in software patents, we estimated two sets of

---

<sup>57</sup> Our analysis is similar to that used in Bronwyn H. Hall & Rosemarie Ham Ziedonis, *The Patent Paradox Revisited: An Empirical Study of Patenting in the U.S. Semiconductor Industry, 1979–1995*, 32 *Rand J. Econ.* 101 (2001). Following Hall & Ziedonis, *supra*, and Jerry Hausman, Bronwyn H. Hall & Zvi Griliches, *Econometric Models for Count Data with an*

models: one using only software patents and a second using total patents. The sections that follow report our findings on the five explanations discussed above: size; share of revenues from product licensing; R&D intensity; share of revenues from hardware sales; and industry sector.

---

Application to the Patents R&D Relationship, 52 *Econometrica* 909 (1984), and Bessen and Hunt, *supra* note 55, we estimate our models using both Poisson and negative binomial distributional assumptions.

Similar to previous work we assume that the number of patents applied for in a year is a function of a firm's R&D spending and other characteristics of the firm. The subscript  $i$  denotes the firm, and the subscript  $t$  denotes the year. The number of patents produced by firm  $i$  at time  $t$  is denoted by the variable  $y_{it}$ . We assume that the number of patents is a function of observable and unobservable factors. The primary estimates in this paper assume that the unobserved component has a Poisson distribution. Under the Poisson distribution assumption the expectation of  $y_{it}$  takes the form:

$$E(y_{it}) = \exp(x_{it}\beta)$$

The expectation of the model is a function of observed exogenous variables  $x_{it}$  and a vector of parameters  $\beta$ . The parameters of the model are estimated using maximum likelihood. We note here an important feature of our analysis. In general, a maximum likelihood model will not be consistent unless the distributional assumption of the model is correct. However, Christian S. Gourieroux, Alain Montfort & Alain Trognon, *Pseudo Maximum Likelihood Methods: Application to Poisson Models*, 52 *Econometrica* 701 (1984) show that if the mean of the above equation is correctly specified then the estimate of  $\beta$  will be consistent even if the data rejects the Poisson distributional assumption. The standard errors must be corrected to be robust to alternative distributions. This is important because the assumption that the variance of the Poisson model is equal to the mean is restrictive and often (as with the data here) incorrect in practice, typically when the excess of the variance over the mean reflects "over-dispersion." We discuss in detail below how we have addressed the problems in matching the distributional assumptions of those models to the characteristics of this dataset.



B. Analysis

1. Patenting and Size

As suggested above, it is natural to expect that patenting would correlate with size to some degree. This might be true because of economies of scale in patenting, it might be true because larger firms are more likely to have matured to the stage where they can develop sophisticated patenting policies, or it might be true because larger firms are more likely to derive value from patents than smaller firms. Or, it might simply relate to litigation exposure, as discussed above.

Tables Two and Three summarize our analysis. Table Two presents the software patent model, and Table Three presents the total patent model. We report t-statistics in parentheses after the coefficient. Given its intuitive appeal, it is not surprising that size is relevant, measured by  $\log(\text{Employee})$ .<sup>58</sup> That variable has a stable sign (positive) and a stable magnitude (slightly more than one in all four models), and is statistically and economically significant in each of the four models. Because the variable is a log transformation of the raw employee data, the coefficient should be interpreted as a constant elasticity. For instance, all other things held fixed, the coefficient of 1.17 in Column 1 of Table Two on  $\log(\text{Employee})$  implies that a 10% increase in the number of employees causes an 11.7% increase in the number of software patents.<sup>59</sup> Because this

---

<sup>58</sup> Because of the skewed distribution we regress  $\log(\text{Employee})$  rather than the raw data on number of employees.

<sup>59</sup> This result is slightly higher, but comparable to other results found in the literature, including Hall and Ziedonis's calculation of a coefficient of .989 in the semi-conductor industry. See Hall & Ziedonis, *supra* note 57. It also is slightly higher than Bessen and Hunt's calculation of a coefficient of .88 in the production of software patents by firms that are for the most part outside the software industry. See Bessen and Hunt, *supra* note 55.

coefficient is slightly more than one, it suggests that returns to scale in number of employees are approximately constant in the software industry. In other words, other things held equal, firms patent in proportion to their size, so that if the number of employees is doubled, we would expect the firm to produce slightly more (12%) than twice the number of patents.

Table 2: **Propensity to Produce Software Patents**

	(1) Poisson		(2) Negative Binomial		(3) Poisson		(4) Negative Binomial	
Log(R&D/Employee)	<b>0.889</b>	<b>(3.83)</b>	<b>0.828</b>	<b>(3.28)</b>	<b>0.599</b>	<b>(2.56)</b>	<b>0.934</b>	<b>(3.44)</b>
Log(Employee)	<b>1.171</b>	<b>(14.86)</b>	<b>1.094</b>	<b>(13.84)</b>	<b>1.138</b>	<b>(12.07)</b>	<b>1.061</b>	<b>(12.09)</b>
Services	<b>-0.023</b>	<b>-(2.87)</b>	<b>-0.040</b>	<b>-(5.58)</b>	<b>-0.015</b>	<b>-(2.11)</b>	<b>-0.020</b>	<b>-(2.43)</b>
Fraction Software Sales	-0.009	-(1.78)	-0.002	-(0.25)	<b>-0.008</b>	<b>-(2.43)</b>	-0.005	-(0.65)
Zero R&D	<b>-4.770</b>	<b>-(5.11)</b>	<b>-4.259</b>	<b>-(2.33)</b>	<b>-4.020</b>	<b>-(5.07)</b>	<b>-4.907</b>	<b>-(2.38)</b>
Year 1999	0.009	(0.04)	-0.096	-(0.31)	-0.136	-(0.54)	-0.225	-(0.70)
Constant	-3.788	-(2.42)	-3.555	-(2.84)	1.349	-(0.23)	-3.797	-(2.74)
Alpha			4.458	(6.25)			2.921	(5.67)
Sector Fixed Effects	No		No		Yes		Yes	
# Observations	703		703		612		612	
# Firms	511		511		445		445	
log-likelihood	-1049		-500.2		-749.8		-448.7	

Table Three: **Propensity to Produce Patents**

	(1) Poisson		(2) Negative Binomial		(3) Poisson		(4) Negative Binomial	
Log(R&D/Employee)	<b>0.614</b>	<b>(2.31)</b>	<b>0.915</b>	<b>(4.81)</b>	<b>0.681</b>	<b>(2.70)</b>	<b>0.800</b>	<b>(4.53)</b>
Log(Employee)	<b>1.071</b>	<b>(13.88)</b>	<b>1.046</b>	<b>(13.79)</b>	<b>1.143</b>	<b>(14.43)</b>	<b>1.055</b>	<b>(14.41)</b>
Services	<b>-0.026</b>	<b>-(3.80)</b>	<b>-0.033</b>	<b>-(5.27)</b>	<b>-0.015</b>	<b>-(2.28)</b>	<b>-0.025</b>	<b>-(4.63)</b>
Fraction Software Sales	<b>-0.008</b>	<b>-(2.13)</b>	-0.004	-(0.78)	-0.008	-(1.60)	-0.004	-(0.94)
Zero R&D	<b>-4.219</b>	<b>-(4.08)</b>	<b>-5.157</b>	<b>-(6.05)</b>	<b>-4.001</b>	<b>-(3.89)</b>	<b>-4.148</b>	<b>-(4.85)</b>
Year 1999	0.442	(1.49)	0.479	(1.53)	<b>0.527</b>	<b>(2.26)</b>	0.443	(1.87)
Year 2000	-0.121	-(0.38)	-0.179	-(0.53)	0.116	(0.52)	-0.221	-(0.91)
Constant	-3.770	-(3.69)	-2.869	-(3.67)	-5.376	-(4.33)	-4.333	-(4.09)
Alpha			4.187	(5.89)			3.135	(5.89)
Sector Fixed Effects	No		No		Yes		Yes	
# Observations	1105		1105		1029		1029	
# Firms	642		642		589		589	
log-likelihood	-1975		-905.2		-1427		-830	

The size of the coefficient sheds light on the relation between size and patenting practices. For example, if patents were substantially more useful for larger firms, the coefficient would be substantially greater than one.<sup>60</sup> If patents were substantially more useful for smaller firms, the coefficient would be substantially less than one, because the size of the portfolio would not increase as quickly as the size of the firm. Because the coefficient is so close to one, it seems likely that the other variables in our model have captured the size-related reasons why patents have different utilities for firms.

## 2. Products and Services

The most promising explanation for the variation in patenting practices, drawn from the business-school literature, focuses on a continuum from products firms to

<sup>60</sup> Thus, our results are inconsistent with the predictions of Gideon Parchomovsky & R. Polk Wagner, Patent Portfolios, 154 U. Pa. L. Rev. 1, 52-57 (2005).

services firms.<sup>61</sup> Firms that sell software products generally have higher operating margins, higher growth rates, and less stable market shares, whereas services firms generally have lower operating margins and lower growth rates, but can more readily establish stable market positions.<sup>62</sup> Thus, a products firm such as Microsoft will have high-volume sales of non-customized products that customers can use “off the shelf” with little or no assistance, and a typical services firm such as EDS will generate revenues by helping firms to install, design, and maintain software. The products model is relatively more effective for venture-backed startups than the services model. Because products firms can scale more easily than services firms, successful products firms are more likely to produce the high returns venture capital investors seek. There also are a large number of hybrid firms like Oracle. Some of those firms began by attempting to sell products, but later were forced by market conditions to provide increasing levels of customization, thus degrading their ability to sell high volumes of a pure high-margin product. To get a sense for the variation, Figure Seven displays for five of the largest software firms the differing shares of revenues attributable to product licensing.<sup>63</sup>

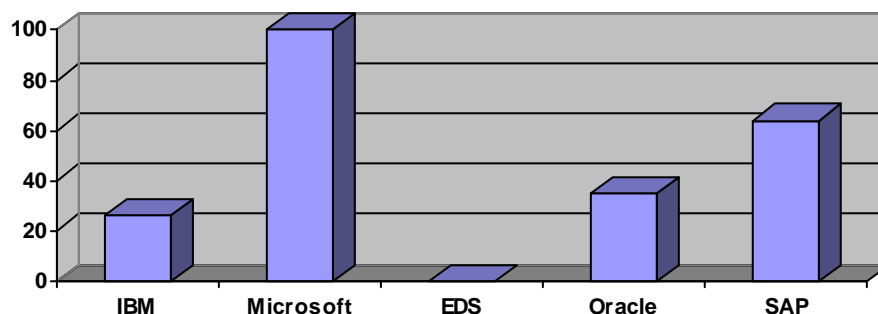
---

<sup>61</sup> See Michael A. Cusumano, *The Business of Software* (2004).

<sup>62</sup> *Id.*

<sup>63</sup> Although accounting practices might differ slightly, product licensing encompasses the fees generated from software products and excludes services revenue related to maintenance, support, consulting, and the like.

**Figure Seven: 2002 Products Revenue Share for Leading Software Firms**



Given the obvious difference in appropriation strategies, the products-services distinction provides a useful lens for exploring the apparent disparity of patenting practices in the industry. Patents seem likely to be a relatively more effective tool for protecting innovation in products than in services. To the extent a firm can provide a unique level of skilled services, it may be feasible to maintain much of the differentiating knowledge in a tacit form, bound up with the skills of the individual employees. Conversely, a products firm that sends its product out into the marketplace in many instances will be vulnerable to appropriation by competitors.<sup>64</sup> If so, a patent that permits a firm to fence out competitors will have considerably more value to a products firm than to a services firm. This, in turn, suggests the hypothesis that products firms, because their technology is more difficult to protect than the technology of services firms, will produce more patents than services firms, all other things being equal.

Because the *Services* variable is a fraction of software revenues, and because our hypothesis is that the devotion of the firm to a products model should relate positively to

---

<sup>64</sup> See Pamela Samuelson, Randall Davis, Mitchell D. Kapor & J.H. Reichman, A Manifesto Concerning the Legal Protection of Computer Programs, 94 Colum. L. Rev. 2308 (1994); Mann, Software Patents, *supra* note 1.

the firm's propensity to patent innovations related to software, the results in Table Two should provide the clearest test of our primary hypothesis, with Table Three primarily relevant as a robustness check. In Table Two, the *Services* variable is negative and significantly related to the dependent variable in all of the different runs. The impact of the *Services* variable on the number of patents also appears to be economically significant. For example referring to the base (Poisson) model in column 1, the coefficient suggests that a 1% increase in the percent of software sales coming from services (e.g. percent of sales increasing from 50% to 51%), implies a 2.3% decrease in the number of patents produced.<sup>65</sup> A more extreme result suggests that the magnitude of the *Services* variable is also economically significant. A firm that derives all its revenues from products (e.g. *Service*=0%) is expected to produce 230% more patents than a firm entirely devoted to providing services (e.g. *Service*=100%).

Those findings are robust. For example, the sign and general magnitude of the coefficient were stable in a model (reported in Column 2 of Table Two) using a negative binomial distributional assumption instead of our base Poisson assumption.<sup>66</sup> The

---

<sup>65</sup> Although the goodness of fit test rejects the Poisson distributional assumption, we nevertheless report the results of this analysis, following prior statistical practice. See Gourerieroux, Monfront & Trognon, *supra* note 57. As recommended there, we use heteroscedastic-consistent standard errors to calculate t-statistics. The goodness of fit test is based on the deviance statistic. The standard error estimates used to compute the t-statistics are robust to heteroskedasticity and misspecification of the distribution. To account for the multiple observations of some firms and the consequent possibility of autocorrelation, the standard errors are clustered.

<sup>66</sup> We use a negative binomial model because the goodness of fit test suggested that our data is overdispersed. The negative binomial model is consistent only if the true distribution is negative binomial; however, if this is the true specification then the estimate is more efficient

parallel runs using total patents as the dependent variable, reported in Table Three, are similar.<sup>67</sup> Most importantly (as summarized in Columns 3 and 4 of Table Two), we also estimated a series of models that include sector-specific fixed effects to control for differences in the propensity to patent across different sectors in the software industry.<sup>68</sup> Inclusion of the sector fixed effects is important for two reasons.<sup>69</sup> First, the estimates in Columns 1 and 2 do not indicate whether the services variable is capturing different propensities to patent that relate to differences between sectors, or whether the product/services distinction is also important within sectors. Inclusion of the sector fixed effects along with the service variable focuses the test of the service variable.

---

than the Poisson model. Referring to Column 2 of Table 2, the parameter alpha is the overdispersion parameter. The high T-statistic, indicating that alpha is significantly different from zero, indicates that overdispersion remains even in the negative binomial model.

<sup>67</sup> This has the advantage of having more data points (because we can use the additional 20,000 patents from 2003-05), but it has the disadvantage that we must analyze total patents rather than software patents (because we have not divided the later patents into software and non-software patents). In any event, those runs produced results and coefficients similar to those set out in Table 2.

<sup>68</sup> We test the joint statistical significance of the sector fixed effects by using a likelihood ratio test based on the selected sample. For both models, we reject the null hypothesis that the sector specific fixed effects have no explanatory power at the 95% confidence level.

<sup>69</sup> Inclusion of sector specific fixed effects necessitates dropping several observations from the analysis. Sectors that have no patents are excluded from the analysis because the sector specific fixed effects entirely explain the number of patents in those sectors. In addition, the sector category marked “Other” is also excluded because it does not represent any particular sector. The sector fixed effect estimates are based on the remaining 612 observations from the 445 remaining firms.

Specifically, this model shows that the product/services distinction is important within sectors.

The results of the Poisson model and the negative binomial model both indicate that the *Services* variable continues to be negative as well as statistically and economically significant. What this suggests is that the devotion of a firm to a products or services model is important, even within a particular sector. Thus, the data do not suggest simply that some sectors of the industry rely more on products and some more on services or that those differences can explain levels of patenting activity. Rather, the data suggest important differences along the products/services continuum, even within particular sectors. To be sure, the magnitude of the coefficients on *Services* does drop considerably (from .023 and .040 to .015 and .020, respectively), but this merely suggests that sector differences capture a portion of the difference in patenting activity.

We emphasize the ambiguity of causation in this finding. On the one hand, it well might be that firms have a higher propensity to patent because they are products firms, either because of the greater ability of those patents to protect innovation in products or because of the greater need to protect innovation that is disclosed through the distribution of products. Conversely, however, it well might be that a patenting culture in a firm's early days could contribute to its survival as a products firm. Referring back to Figures One and Two, the most dominant products firms (Microsoft and Adobe, with 100% and 99% products revenue shares) were among the earliest prepackaged software firms to start patenting, while the three late patentees shown in those Figures survive today as the most prominent hybrid firm (Oracle, with a 35% products revenue share) and as two of the largest services firms (Autodesk and Computer Associates, with 0% products revenue



shares). It is at least possible to believe that the late patenting strategies of those firms undermined their competitive position in products markets and forced their retreat into the realm of non-products firms.

### 3. Patenting and R&D Intensity

The next possibility is that a firm's R&D intensity affects its production of patents. Specifically, assuming that the firm has the same number of employees, the same share of products/services revenues, and is in the same sector in the industry, will the number of patents relate to the intensity of R&D investment (measured by dollars of R&D investment per employee)? The discussion in Part II regarding the increase in the number of patents as a function of R&D strongly suggests that R&D intensity is important. Not surprisingly, Tables Two and Three suggest that R&D intensity does explain an important portion of the variation in patenting rates.

Like the previous variables, R&D intensity has a stable sign (positive) and a coefficient that is both statistically and economically significant. Specifically, the elasticity of R&D intensity on patenting is .89. Again, this is similar to the results found by Bessen and Hunt of 1.01.<sup>70</sup> However, these estimates are much larger than results in the semiconductor industry of .18 found by Hall & Ziedonis. Generally, this suggests that although the effects of size in the software industry are about the same as those in the

---

<sup>70</sup> Bessen and Hunt, *supra* note 55, find different results when accounting for firm level heterogeneity using fixed effects. However, their fixed effect estimation excludes firms with zero patents. Such firms include a majority of firms in this paper and in the Bessen and Hunt paper. Therefore, we compare the basic Poisson regressions as these include all firms and are less prone to sample selection bias.

semi-conductor industry, the effect of R&D intensity on software patenting is quite a bit greater than its effect on semiconductor patenting.

#### 4. Patenting and Hardware Revenues

Another possibility we attempted to test is that hardware or systems firms will have different patenting strategies than pure software firms. This hypothesis is drawn from the common suggestion in interviews with software executives that the typical hardware firm has a culture that is different from that of the typical software firm.<sup>71</sup> The typical hardware firm culture emphasizes institutions that facilitate patenting; the typical software firm culture resists those institutions. Because the hypothesis here relates to the overall patenting philosophy of the firm, the hypothesis is tested more directly with the data on total patents in Table Three.

The results here are markedly less robust than for the other variables. The relevant variable for this question is *Fraction of Software Sales*. As Table Three displays, the results range from marginal statistical significance (in the Poisson model) to no significance in the other models. The coefficient also is quite small, though usually with the negative sign that the hypothesis suggests (indicating that firms with higher shares of revenues from software rather than other lines of business are likely to have fewer patents than firms with lower shares of revenue from software sales).<sup>72</sup>

---

<sup>71</sup> See Mann, *supra* note 52, at 980 n.102.

<sup>72</sup> We also have run a number of robustness checks. These include not only the Table Two models analyzing software patenting rates (rather than total patenting rates), but also runs that include a number of outliers with very large portfolios, a random effect Poisson estimate with sector-specific fixed effects, and firm-level fixed effects. Those results are similar to those we report in the text.

What is most interesting about those results is the small coefficient, which suggests that any distinction between the patenting practices of pure software firms and firms with substantial non-software revenues is slight. In light of the history summarized in Part II, this suggests that by the time our data were collected, the cultural resistance to patenting in the software industry had lost much of its force. Even if there is a slight lingering distinction, patenting already had become as routine for software firms as it long has been for hardware firms.

#### 5. Patenting and Software Sectors

The final variable we considered is the relation between industry sectors and patenting propensities. As discussed above, differences among the widely heterogeneous sectors in the software industry may explain some of the differences in patenting practices between firms. The question remains, in light of the sections above, whether our more general variables capture the reasons for patenting variations between sectors. The problem in using the Software 500 sector designations is that Software Magazine during the period for which we collected data has used more than 100 different sector designations, many of which include very few firms. Accordingly, we constructed a modified set of sectoral designations, which consolidates the Software 500's designations into "only" 36 sectors.<sup>73</sup> Table Four below lists the different sectors and provides basic descriptive statistics for the firms in the sector.

---

<sup>73</sup> A good deal of our consolidation reflected collapsing different designations used from year-to-year for similar firms.

# Software Patents, Incumbents, and Entry

**Table Four: Software Sector Descriptive Statistics (1998 and 1999)**

Representative F	R&D	Empl. (1000s)	R&D per Emp	Services	SW Patents	SW Patents	SW Patents	SW Patents	Count of Obs.
	Median	Median	Median	Median	Mean	Median	75%	90%	
Application Dev'l/pm't	6.95	220	0.032	22	0.23	0	0	1	61
Application Serv. Prov.	1.39	69.5	0.017	26	1.75	0	3.5	7	4
Asset/Technol. Mgmt.	7.38	421.5	0.027	38.5	0.00	0	0	0	12
Business Intelligence	12.83	446	0.029	28	0.37	0	1	1	27
Bus. Process Mgmt.	21.51	525	0.035	54	0.33	0	1	1	3
Computer Ass'd Draft'g	5.72	283	0.020	32	1.57	0	0	11	7
Content/Doc. Mgmt.	8.71	188.5	0.031	18	0.18	0	0	0	28
Collab./Proj. Mgmt.	10.40	376	0.029	38	0.00	0	0	0	4
Cust. Rel'nship Mgmt.	5.84	206	0.031	35	0.32	0	0	0	41
Database	18.84	505.5	0.032	25	9.69	0	2	62	16
Disaster Recovery	0.84	38	0.022	27	0.00	0	0	0	1
Data Warehouse	298.02	5679	0.044	12	15.29	0	49	57	7
Enterprise Appl. Integr'n	6.48	261	0.030	30.5	0.06	0	0	0	36
E-Business Applications	7.27	217.5	0.027	31	0.25	0	0	1	44
E-Commerce	13.57	195	0.032	13	0.00	0	0	0	5
E-Learning	34.18	1500	0.023	26	0.00	0	0	0	3
Enterprise Res. Planning	11.25	522.5	0.023	40.5	0.10	0	0	0	62
Financial Applications	8.29	270	0.027	32	0.02	0	0	0	42
Geogr. Info. Systems	18.05	420	0.029	9	6.00	0	9	32	7
Healthcare	3.33	106	0.020	22	0.00	0	0	0	9
Human Resources	3.05	31	0.022	26.5	0.00	0	0	0	16
Infrastructure	7.85	250	0.037	23	1.33	0	0	2	55
IT Sourcing	5.73	300	0.021	53	40.15	0	0	138	13
Marketing Automation	2.23	78	0.024	23	0.00	0	0	0	5
Middleware	5.67	200	0.029	17	0.42	0	1	2	19
Operating Systems	25.71	641	0.039	14.5	43.14	1	2	271	14
Portal Tools	3.32	151.5	0.026	34.5	0.00	0	0	0	8
Publishing/Graphics	9.12	300	0.034	0	3.20	1	4	11	5
Retail Applications	27.74	1068	0.026	74	0.00	0	0	0	1
Supply Chain	8.25	253	0.030	42.5	0.63	0	0	1	54
Security	12.11	450	0.032	13.5	1.50	0	1.5	6.5	20
Sales Force Automation	7.21	381	0.039	39.5	0.00	0	0	0	4
System Integration Servs.	9.53	2240	0.011	49	65.00	0	35.5	361	16
Storage Management	3.79	135	0.032	11	7.14	0	4	43	7
Vertical Indus. Appl.	3.57	273.5	0.015	39.5	1.60	0	0	0	20
Wireless/Mobile	3.33	119.5	0.033	37	0.25	0	0.5	1	4
Other	6.04	164	0.031	20	1.65	0	0	2	23
Total # of Firms:									

The variation in the median of size and service revenue underscores the heterogeneity of the sectors. For example, the median data warehousing firm has about 5,000 employees, while the median disaster recovery firm has only 38. Similarly, the typical data warehousing firm derives only 12% of its revenues from services, while the median retail applications firm derives 74% of its revenues from services.

Of particular import for our work is the variation in patenting practices, with quite a number of reasonably well-populated sectors entirely devoid of patents (human

resources software, for example), and others in which substantial portfolios exist (operating systems and systems integration services, for example, with an average of more than 40 patents per firm). Table Five presents a more rigorous examination of that question – a sector fixed effect analysis designed to illustrate the particular effects on patenting propensity of each of the sectors. As you would expect given the discussion above, Table Five illustrates stark differences among sectors that are not captured by the other variables.

Perhaps the most interesting point from that table is the apparent relation between concentration in a particular sector and patenting propensity. As Figure Eight illustrates, the sectors with highest propensities to patent have fewer firms per sector than the sectors with moderate or low propensities to patent. It is difficult to be sure why industry concentration would relate to patenting propensity. One possibility suggested by Cockburn and MacGarvie<sup>74</sup> is that the presence of substantial patent portfolios may deter further entry into the sector. Another possibility is that weaker firms disappear as sectors mature. If substantial portfolios are a feature of relatively mature firms, we would expect mature sectors to have a smaller number of firms with greater average rates of patenting. The breakdown of sectors in Table Five provides some support for this possibility.

---

<sup>74</sup> See Cockburn & MacGarvie, *supra* note 7. One problem with that explanation, at least with respect to Cockburn & MacGarvie's data, is that they analyze entry in any given year as a function of existing patent portfolios. But most venture-backed software startups do not obtain patents until after several years of operations. Accordingly, a sector in which several firms already have strong portfolios is likely to be a sector of relatively mature technology. It should be no surprise that the rate of entry will slow in such a sector, but it is just as likely attributable to the head-start and first-mover advantages of the existing firms as it is to the exclusive force of the patents held by the existing firms.

**Figure Eight: Patenting Rates and Sector Concentration**

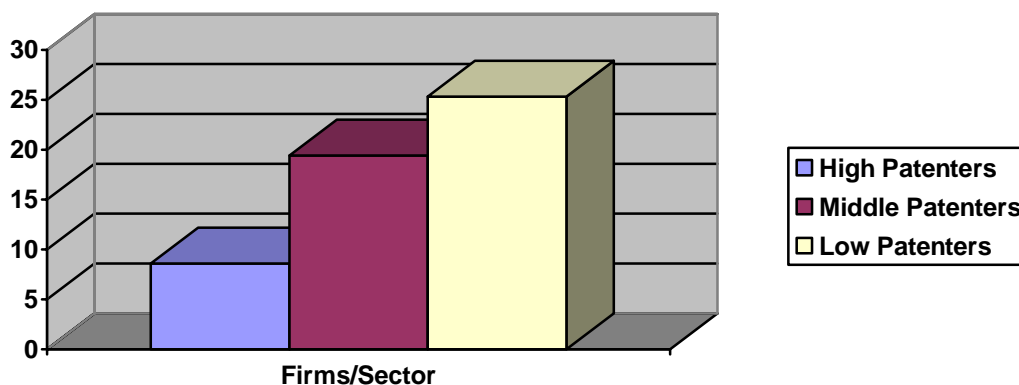


Table Five: **Sector Fixed Effect Analysis**

Software Sector		# Firms # Obs		Poisson Model			Neg. Binomial		
				Fixed Effect	S.E.	Rank	Fixed Effect	S.E.	Rank
VA	Vertical Indus. Appl.	15	20	1.925	(0.82)	1	0.841	(0.85)	12
ASP	Application Serv. Prov.	4	4	1.883	(0.45)	2	1.464	(0.68)	2
GIS	Geogr. Info. Systems	5	7	1.607	(0.69)	3	1.288	(0.81)	5
OS	Operating Systems	10	14	1.591	(0.75)	4	1.071	(0.65)	4
PU	Publishing/Graphics	5	5	1.473	(0.58)	5	1.435	(0.79)	3
SEC	Security	15	20	1.392	(0.77)	6	1.492	(0.94)	6
DB	Database	12	16	1.215	(0.42)	7	0.785	(0.68)	10
WVM	Wireless/Mobile	3	4	1.202	(1.24)	8	1.686	(1.22)	8
SC	Supply Chain	38	54	1.159	(0.70)	9	0.435	(0.67)	17
CAD	Computer Ass'd Draft'g	5	7	1.115	(0.74)	10	0.636	(0.82)	16
CRM	Cust. Rel'nship Mgmt.	32	41	1.105	(0.74)	11	0.651	(0.69)	14
ITS	IT Sourcing	9	13	1.030	(0.65)	12	0.816	(0.70)	9
SIS	System Integration Servs.	10	16	0.856	(0.73)	13	1.123	(0.77)	7
SM	Storage Management	5	7	0.747	(0.72)	14	3.518	(1.20)	1
MW	Middleware	16	19	0.686	(0.51)	15	0.940	(0.85)	11
BI	Business Intelligence	20	27	0.407	(0.58)	16	0.665	(0.73)	15
INF	Infrastructure	39	55	0.391	(0.73)	17	0.623	(0.64)	13
DW	Data Warehouse	5	7	0.378	(0.61)	18	0.596	(0.92)	18
BPM	Bus. Process Mgmt.	3	3	0.369	(0.47)	19	0.128	(0.68)	21
CDM	Content/Doc. Mgmt.	22	28	0.055	(0.95)	20	0.253	(1.06)	20
AD	Application Dev't/pm't	42	61	0.000	-	21	0.000	-	22
EAI	Enterprise Appl. Integr'n	26	36	-0.671	(0.84)	22	-0.576	(1.00)	23
EB	E-Business Applications	31	44	-0.852	(1.09)	23	0.472	(0.78)	19
ERP	Enterprise Res. Planning	44	62	-1.409	(0.72)	24	-0.898	(0.91)	24
FI	Financial Applications	29	42	-2.258	(1.14)	25	-1.930	(1.41)	25

\*\*\*\*\*

Fitting this together, three points warrant emphasis. First, although patenting does relate to the size of the firm, the pattern is complex with other variables explaining important parts of the picture. Second, the variables that are most successful in explaining patenting variations are the variables that explain the firm's particular niche

within the software industry; the extent to which a firm has non-software business lines is not important. Third, although we have not emphasized it above, it is interesting that the results in Table Two are so similar to the results in Table Three. Given the traditional view that software patents are much less effective at appropriating the value of innovation than hardware patents, we would have expected the patent production functions for software patents and total patents to differ substantially. The similarity of those functions suggests, again, that the use of software patents is converging rapidly with the use of patents in adjacent sectors.

#### **IV. THE ROLE OF PATENTS IN SOFTWARE DEVELOPMENT**

A complete picture of the software industry cannot be limited to incumbent firms. As others have recognized, the industry experiences high rates of new entry and turnover even at the highest levels.<sup>75</sup> In large part, this is because a great deal of the new technology in the industry is developed not by incumbent firms in the first instance, but rather in one of three development channels that are distinct from the incumbent firms: venture-backed firms, open-source development, and independent inventors.<sup>76</sup> As they do for the incumbent firms, patents play a distinct role in the success of firms in each of those channels.

---

<sup>75</sup> Merges, *supra* note 7. For quantitative analysis, see Cockburn & MacGarvie, *supra* note 7.

<sup>76</sup> The channels are not mutually exclusive. For example, there are a number of open source venture-backed startups. And, venture-backed startups obviously are similar to independent inventors in many respects.

A. The Venture-Backed Startup Channel

The first and most prominent channel is the venture-backed startup channel.<sup>77</sup> In this channel, venture capital firms serve as investment intermediaries, providing capital and management expertise to young firms seeking to make their way into the industry. This channel, of course, has produced many of the most prominent success stories, with Google providing the most salient recent example. As mentioned in Part III, venture financing tends to favor products firms, primarily because successful products firms are much more capable of scaling quickly than services firms.<sup>78</sup> Because the quick ability to scale relates closely to the ability of venture capitalists to obtain the return they seek on their investments, this model tends to work much better for products firms.

Patents do play an important role for startup firms. As one of us has explained previously, patents provide little benefit to the early-stage pre-revenue startup firm. But as the firm matures and begins to develop revenue streams, patents become increasingly important. This can be because the firm needs the patents to prevent larger firms from copying its products. It can be because the patents will be important to investors as the firm's financing needs increase. Or it might be because the patent signals something about the firm's sophistication or management acumen. But whatever the reason, interviews with investors and entrepreneurs strongly suggest that patents can be important for venture-backed software startups. The perspective is not unanimous.

---

<sup>77</sup> This section draws heavily on Mann & Sager, *supra* note 7.

<sup>78</sup> For the same reason, computer and peripheral firms are not as readily suited to venture capital investment. For example, the PricewaterhouseCoopers/National Venture Capital Association MoneyTree Report lists 869 software investments for \$4.8 billion for 2005, but only 62 investments in computer and peripheral firms, totaling only \$500 million for the same year.



Many, if not most, investors recognize the limited value that patents have for appropriating the value of a software innovation, and thus worry about how their portfolio firms will defend a market share even if they can develop a significant product.

That perspective is buttressed by the available data about the role of patents in venture-backed startups. Although only about 25% of venture-backed firms obtain patents,<sup>79</sup> there is a close relation between the acquisition of patents and the progress of firms through the venture-capital cycle. Thus, for example, firms with patents are likely to obtain more financing and they are more likely to succeed.<sup>80</sup> In general, the relevance of patents to mature and successful portfolio firms is consistent with the discussion at the end of Part II. As these firms mature, patenting becomes a routine part of the firm's operations, just as it has been for many software incumbents for the last decade.

For our purposes, the exit strategy is what is most important about the venture-backed startup channel. Generally, the venture-backed startup that develops a successful product will exit from its startup status in one of two ways. First, it might obtain sufficient funds (generally from public investors) to become a large firm, ascending to the ranks of the incumbent firms in the industry (like Google). Alternatively (perhaps much

---

<sup>79</sup> The share of firms with patents in this venture-backed dataset is higher than the share of firms with patents in the Software 500 dataset described in Part III. This is true, presumably because the venture-backed dataset includes a smaller share of services firms.

<sup>80</sup> With respect to financing, firms with patents obtain a median of four rounds rather than three, worth \$26 million rather than \$15 million. Moreover, within five years of first financing, 13% of the firms with patents go public and only 4% fail. Conversely, 3% of the firms without patents will have gone public by that point in time and 8% will fail. As reported in more detail in Mann & Sager, *supra* note 7, all of those distinctions are statistically significant and stable across a series of checks for robustness.

more commonly), it might sell its technology (normally through a sale of the entire firm) to one of the large incumbent firms.

What is most unlikely to happen is litigation – rare is the startup firm that exploits its technology through patent litigation against a large incumbent firm.<sup>81</sup> Although economic theory suggests that it often is difficult for one firm to transfer valuable information to another (Arrow's information paradox), the venture-backed channel avoids that difficulty. Most obviously, the parties in control of mature venture-backed portfolio firms are a small class of venture capitalists, often previously employed at large incumbent firms. Thus, it is easy to expect that those people would have personal relations that would enhance their ability to make credible representations about technology and come to consensual arrangements for acquisition of the startup. Similarly, the entrepreneurs themselves, to the extent they have any control over the process, are likely to be repeat players, worried about future transactions, and also former employees of incumbent firms themselves. It is easy to see why successful startups often are acquired by incumbent firms and why litigation to enforce IP against incumbent firms is rarely the chosen strategy.

#### B. The Open Source Channel

The open-source channel has gained prominence in recent years as programs like Linux, Apache, and Firefox have been broadly accepted.<sup>82</sup> With its roots in the hacker mythology of the early days of the Internet, the open-source community venerates a

---

<sup>81</sup> Licensing of technology from startups does happen occasionally, though it is rarely the preferred business model of the venture capitalist. Rather, it is a strategy to which the firm turns when it is unable to execute its chosen path.

<sup>82</sup> This section draws heavily on Mann, *supra* note 18.

decentralized style of software development that is the antithesis of the large bureaucratized incumbent firm.

In the last decade, the open-source community has undergone a profound change. At the same time as its products have become sufficiently successful to gain widespread use in large enterprises, the community's development processes have been adopted by some large incumbent technology firms – a few from the software industry (like IBM and Novell), but mostly from adjacent hardware industries (firms like Intel, HP, and Fujitsu). Thus, the commercially successful open-source programs share the salient characteristic that they benefit from extensive financial support from large incumbent firms. The firms making those investments have done so as part of a “value-chain” strategy, in which the firms seek to commoditize a part of a value chain in which they are unlikely to dominate (like the operating system), hoping to extract value at some other part of a value chain (like the servers on which the operating system runs, the middleware that runs on the stack above the operating system, or the services necessary to assemble all of those pieces into a well-designed “solution”).

As the software has become commercialized, an increasing number of purely open source firms have appeared. For the most part, the largest of these firms depend on sales of services. Because open source software can be copied and sold freely by competitors, it is difficult to profit directly from product licensing. Accordingly, service companies dominate this market. The recent battle between Oracle and Red Hat illustrates this point – Oracle apparently plans to copy Red Hat's version of Linux, presumably so that Oracle can profit from licensing products and providing services

related to Linux installations.<sup>83</sup> The relative disutility of patents for services firms together with the communitarian philosophy discussed above makes it unsurprising that there are few patents held by open source firms.

Still, patents and other IP rights are crucial to the success of those strategies in several ways. Most obviously, the incumbent firms use IP to protect their positions in those parts of the value chain where they plan to compete. Thus, for example, although IBM has participated generously in the development of Linux and Apache, and has given the community ready access to the patents relevant to those projects, it has not abandoned the IP strategy that protects its investment in its server lines or software products like Websphere. Also, to the extent that open source communities are protected from patent litigation, it is because they operate under the umbrella of implicit promises of protection from the large patent-holding incumbents that support their communities.

To be sure, major parts of the open-source community find these developments unsatisfying. The Free Software Foundation (FSF) led by Richard Stallman and Eben Moglen, for example, regularly decries the vice of patenting software.<sup>84</sup> Its revisions of the GPL – the license under which Linux currently is distributed – reflect a continuing hostility to the increasing role that patents are playing in the industry. But on this point it is increasingly clear that the FSF no longer speaks for the community as a whole. Thus, we see that Linus Torvalds has expressed great dissatisfaction with the FSF's position on these issues. This is because Torvalds is more concerned about wide deployment and use

---

<sup>83</sup> For discussion of the Oracle strategy, see, e.g., Stephen Shankland, Oracle's Red Hat salvo falls short of mark, CNET News, Oct. 27, 2006.

<sup>84</sup> See Brief of Electronic Frontier Foundation in Support of Petitioner, KSR International v. Teleflex Inc., No. 04-1350 (cert. granted June 26, 2006).

of Linux. Patent-related revisions to the GPL that make it less congenial to the large firms that participate in the Linux project are in tension with that concern.

C. The Independent Inventor Channel

The third channel is the independent inventor. This channel is the most controversial and least susceptible to generalization. Here, we make three points. First, this channel arguably plays a distinct role in providing valuable innovation in the industry. Second, independent inventors are likely to struggle more than other potential new entrants in transferring technology to incumbent firms. Third, building on the first two points, the market response is the creation of intermediaries to facilitate technology transfers between inventors and incumbents. Although the business models of the intermediaries are diverse, they generally describe themselves as patent acquisition or management firms and have been labeled pejoratively as “trolls.” Generally, those firms exhibit a variety of different strategies that respond to the various shortcomings that will hinder independent inventors attempting to exploit their software-related inventions.

1. *The Role of Independent Inventors*

The first point is a relatively subjective one, though none the less significant. Although often vilified in the media as a novel and radical phenomenon, Part II documents the important challenge that independent inventors have presented to incumbents from the earliest days of the industry. Indeed, despite the rhetoric that characterizes the “troll” as an artifact of the rise of the Federal Circuit and related recent events, the inventor that received what often is regarded as the first software patent, more than three decades ago, used his early patents to bolster efforts to create one of the first

software products to enter into serious competition with the “free software” IBM then was bundling with its mainframe computers.<sup>85</sup>

In the current milieu, industry sources (both in large firms and in patent acquisition firms) accept the notion that independent inventors in the software industry often have focused on larger “big-picture” inventions while inventors at incumbent firms have largely focused on incremental improvements to existing product lines. Even at firms like Microsoft, with a corporate culture consciously directed toward forward-thinking innovation, the ever-urgent need to protect and upgrade the firm’s core product lines makes it hard for researchers to do truly basic research about products that cannot be deployed in the near term.<sup>86</sup>

Similarly, venture-backed startups necessarily have a short time horizon because the financing model contemplates success or failure in a *relatively* short time – typically less than a decade. That financing model may be excellent for certain types of innovations, but the lesson of the discussion of venture-backed startups in the previous part is that the constraints of the venture-capital model leave many valuable research opportunities unfunded.

At first glance, it might seem hard to put much weight on the distinction between independent invention and open-source development – apparently the essence of independence. And it surely is true that a grass-roots strain of open-source development persists, evidenced by the thousands of small independent projects registered at

---

<sup>85</sup> See Goetz, *supra* note 13, at 50-53.

<sup>86</sup> Henry Chesbrough’s book *Open Innovation* (2003) provides detailed and perceptive documentation of the difficulties that incumbent high-tech firms have faced in their efforts to foster successfully innovative environments within the boundaries of their own companies.

SourceForge. But historically the path to market traction (and funding) for open-source technology has been distinct from the independent inventor channel discussed in this section. As discussed above, the open-source projects that have gained substantial market traction ordinarily have succeeded through their adoption directly into the value chains of large incumbent firms. It also is fair to say that the key to open-source success has been quality of execution coupled with easy interoperability. Linux and Apache were not visionary advances; they were high-quality solutions to pressing and immediate programming needs. In general, then, open-source development has not (so far at least) provided the path-breaking advances at which independent inventors aim.

## *2. Difficulties of Commercialization for Independent Inventors*

Several overlapping structural considerations make it natural to expect that independent inventors might make valuable discoveries in the industry yet face substantial obstacles that complicate their efforts to commercialize their inventions. The first is the likelihood for many discoveries that direct exploitation by the inventor will be suboptimal, if not wholly impractical. The point is yet another variation on the problem mentioned above – the uncomfortable mapping of dozens (if not hundreds) of inventions into the thousands (or millions) of lines of code in a single software product. An independent inventor could not practicably commercialize an invention that improves Internet browsers, even if the invention is pathbreaking. Thus, the network effects that entrench existing products suggest that the optimal way to deploy a new invention related to Internet browsers is to sell it to one of the incumbent browser developers so that it can be incorporated into their product. As a matter of industry structure, that means that the independent software inventor is more likely than the independent inventor in other fields

to need the kinds of complex business, financial, and legal competencies that are necessary for successful negotiations with the large incumbent firms in the industry.

At the same time, those that populate the independent inventor channel are *less* likely to be well-placed to conduct such negotiations successfully than innovators in the parallel channels.<sup>87</sup> For example, successful innovators in the venture-backed channel are likely to become incumbent firms themselves or to transfer their technology to incumbent firms in a consensual transaction. The venture capitalists that control the destiny of venture-backed startups are a small group of players, often themselves former executives at incumbent firms, likely to enjoy professional and social relationships with the individuals at large firms that might be interested in the technology. Thus, it would be surprising if they could not agree upon a consensual arrangement for transfer of technology to an incumbent firm with a use for the technology. The rarity of patent infringement litigation between venture-backed startups and incumbent firms underscores the point: despite the oft-expressed concern about runaway patent litigation, we are aware of no lawsuit in the industry in which a venture-backed startup has sued a substantial incumbent firm for patent infringement.

Similarly, products in the open-source channel seem to gain widespread commercial traction only after they have been adopted into the value chain of large incumbent firms. Because the successful adoption typically involves a partial merging of the development community with employees of the incumbent (or incumbents) adopting

---

<sup>87</sup> For this reason, the availability of injunctive relief, the issue contested in *eBay*, is crucially important to the intermediaries in this channel. See, e.g., Brief of Rembrandt IP Management, LLC in Support of Respondent, *eBay v. MercExchange, LLP*, 126 S. Ct. 1837 (2006) (No. 05-130); Brief of United Inventors Association and Technology Licensing Corporation in Support of Respondent, *eBay v. MercExchange, LLP*, 126 S. Ct. 1837 (2006) (No. 05-130).



the technology, the frequency of adversarial dispute resolution is small. In any event, the limited frequency of patenting by open-source communities makes the offensive use of patents a strategy that is not readily available.

Lacking those relationships, independent inventors (here as in other contexts<sup>88</sup>) have been forced to resort to litigation to extract value from their inventions.<sup>89</sup> Thus, as we look at patent litigation in recent years in the software industry, we now see that a substantial share of litigation involves such firms or their subsidiaries.<sup>90</sup> To the economist it might seem odd that negotiations in this channel should fail so frequently. But software technology is not easy to transfer. For one thing, software inventors often

---

<sup>88</sup> John R. Allison et al., *Valuable Patents*, 92 Geo. L.J. 435 (2004) (litigated patents tend to issue to individuals or small firms, and often are transferred between issuance and litigation); Jean O. Lanjouw & Mark A. Schankerman, *Stylized Facts of Patent Litigation: Value, Scope and Ownership* (November 1997) (NBER Working Paper No. W6297) (corporate patent owners appear to be less likely to be in litigation than individual patent owners).

<sup>89</sup> Jack Goldsmith's and Tim Wu's discussion of the troubled development of an online music market provides an instructive parallel. It should have been clear to all concerned parties by 2000 that the efficiencies of online music distribution eventually would compel some method of easy digital distribution of music online. But the brash early technology startups like mp3.com and Napster had no success at all in reaching consensual arrangements with the large media providers. See William W. Fisher, *Promises to Keep: Technology, Law and the Future of Entertainment* (2004). But Steve Jobs, largely because of personal relations that spanned the divide between technology firms and the large media companies, was able to reach agreements in one fell swoop with all of those companies, facilitating both the I-Tunes store and the I-pod. See Jack Goldsmith & Tim Wu, *Who Controls the Internet: Illusions of a Borderless World* 118-21 (2006).

<sup>90</sup> That is not to say that there is never litigation among incumbents – IBM's recent filings against Amazon.com and the high-profile litigation between AT&T and Microsoft underscore the occasional inability of large firms to reach rational settlements of these kinds of disputes. *AT&T v. Microsoft*, 414 F.3d 1366 (Fed. Cir. 2005), cert. granted Oct. 27, 2006.

will have difficulty in persuading potential purchasers of the value of their inventions without disclosing the inventions in some detail. Given the difficulty of protecting the value of the technology even with a well-considered patenting program, independent inventors reasonably might be reluctant to make such disclosures. But a failure to disclose (by hypothesis) will make it harder to persuade a purchaser to pay the “true” value of the technology – a standard instance of Arrow’s information paradox.<sup>91</sup> In the end, where venture-backed firms could use their connections to get a receptive hearing from incumbent firms, independent inventors that claim to have discovered valuable technology are more likely to be dismissed as “kooks.”<sup>92</sup> If they cannot obtain a serious hearing from large firms, their only recourse is to resort to legal coercion.<sup>93</sup>

---

<sup>91</sup> See Ronald J. Mann, Verification Institutions in Financing Transactions, 87 Geo. L.J. 2225, 2267-68 (1999) (discussing Arrow’s information paradox).

<sup>92</sup> For example, despite the general perception in the media (and on Capitol Hill) that NTP’s patent claims related to the Blackberry were unfounded, some industry sources portray Tom Campana (the inventor of the patent in question) as a thinker of great perception and foresight.

<sup>93</sup> One interesting developing battleground in this area is the question of “transparency.” Incumbent firms call for transparency in the ownership of patenting, so that they readily can identify the real parties in interest when patents are issued or transferred. Intermediaries anticipating litigation, however, prefer that their acquisition of patents goes unnoticed. This could be true for a spectrum of reasons of varying legitimacy, ranging from a Lemelson-like desire to allow competitors to become more dependent on a patent *before* revealing its existence to more pedestrian concerns such as a desire to control the forum in which litigation will occur. It is not entirely clear what the best solution is, but it is worth noting that many large companies have their patents held by a separate, non-practicing company that has no assets other than intellectual property.

3. *Patent Enforcement Intermediaries*

In a world of perfect markets, the conditions described above would summon into existence intermediaries specializing in the particular competencies that independent inventors are likely to lack: the ability to enforce patents aggressively against incumbent firms, the ability to raise funds to support the continuing development and exploitation of the technology, and (most important from a social perspective) the ability to facilitate the deployment of the technology by licensing it to the firms best-placed to use it. There can be no doubt that a substantial group of these firms has arisen – some of the most prominent (in alphabetical order) include Acacia Technologies Group, Altitude Capital, Intellectual Ventures, and Rembrandt IP Management.<sup>94</sup> Indeed, if there is anything odd about the situation, it is not that some firms have arisen to fulfill those functions. It is that they have taken so long to appear. Because none of the considerations discussed above explaining the rise of these intermediaries have changed substantially in the last decade, it is not easy to see why they have arisen so rapidly in the last few years alone.<sup>95</sup>

But the activities of those firms illustrate that each of them is pursuing a distinct strategy. The best way to understand those strategies is to recognize that different inventors will fail in commercialization for different reasons. Thus, the optimal exploitation strategies for different technologies will be different, which makes it natural

---

<sup>94</sup> There are many smaller firms that exploit particular technologies. E-Pass Technologies, for example, exists primarily to support the licensing of its patented smart card product.

<sup>95</sup> The pejorative discussion in the concurring opinion of Justice Kennedy in *eBay v. MercExchange, LLP*, 126 S. Ct. 1837 (2006) certainly has contributed to the high visibility of the issue as a policy matter and the simplistic pejorative use of the term “troll” to refer to the wide variety of entities discussed here.

to expect that a range of intermediaries would arise specializing in different strategies. As illustrated in Figure Eight, we organize those strategies along two different dimensions: the type of opportunity that the intermediary acquires and the source of funds on which the intermediary relies.

(a) Acquiring Litigation or Technology?

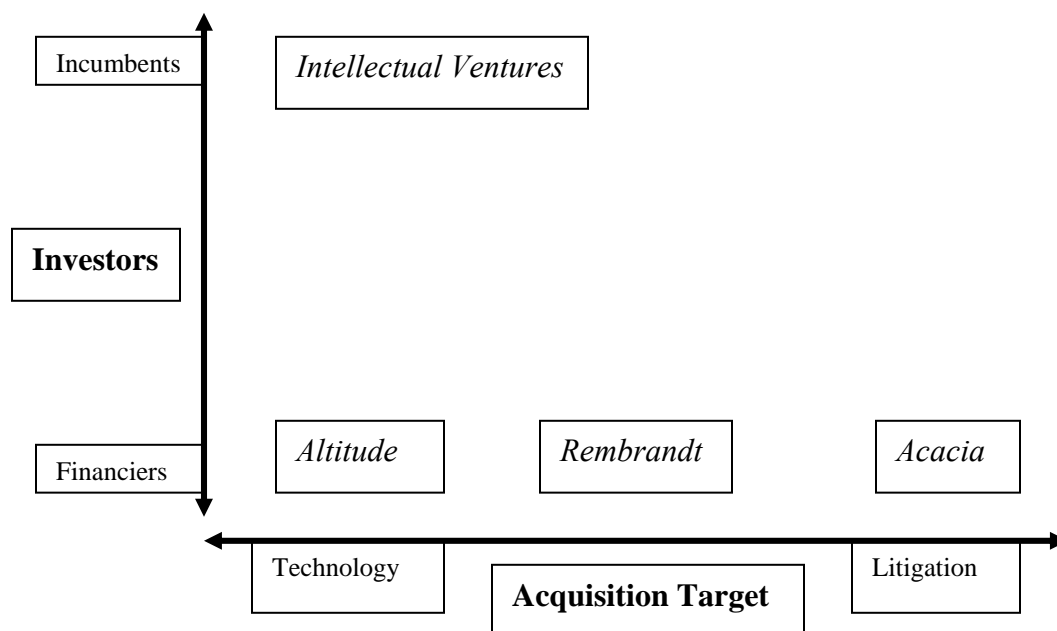
The most fundamental distinction relates to the type of asset on which the intermediary focuses. Here, we discern a spectrum from pure litigation on one end (with relatively little regard for technology) to pure technology at the other end (with little regard for litigation). Thus, at one end of the spectrum we would place firms like Acacia, which function much like the paradigmatic securities class action law firm. Essentially, those firms search for opportunities to acquire patents, frequently from defunct firms.<sup>96</sup> With the patents in hand, they search broadly for companies that might be regarded as infringing those patents, often in market sectors far removed from the market at which the patent originally was directed. Whenever they can present a colorable claim, they should be able to obtain a settlement from the defendant that at least reflects the expected present value of the defendant's litigation costs (what some might pejoratively call a "nuisance" settlement). That activity will be privately profitable whenever those settlements exceed the often trivial cost that the intermediary must pay to acquire the patents. The net social contribution of that activity would depend on the balance between the funds that are flowing back to the original inventor (and thus providing an incentive

---

<sup>96</sup> Acacia in particular has brought suits in several cases already relying on patents purchased from insolvent entities. Because many of those entities are failed venture-backed startups, the activity affords a link through which assets in failed VC startups can be brought into the independent inventor channel. However, it is not clear that the availability of that channel is important to investment decisions.

for the innovative activity that generated the patent), balanced against the resources consumed in the litigation to enforce the patent.

**Figure Nine: Schema of Patent Exploitation Intermediaries**



At the other end of the spectrum are firms focused more on the acquisition of technology than litigation. Here we place a firm like Altitude Capital, a firm with a relatively large share of veterans of hedge funds and investment banks. In general, the goal here is to earn a profit by skillful balancing of present and future cash flows related to the technology. Thus, a typical transaction for Altitude would involve the present acquisition of a patent in return for an immediate payment to the inventor, which Altitude would hope to recoup over time by revenues to be obtained from licensing the patent to an incumbent with a use for the technology. For that transaction to succeed, Altitude must accurately assess the value of the technology – the likely future revenues that the patent will generate – and also must convert those revenues to an anticipatory payment

stream accruing to the inventor. If it is skilled at both of those tasks, Altitude should be able to earn a good return on the funds that it invests. The social value of the activity, again, would turn on the balance between the funds flowing back to the original inventor and the transaction costs associated with Altitude's activity. If Altitude can succeed in generating sufficient revenue streams to earn a profit while both avoiding litigation and returning a substantial stream of funds to the original inventors, it is hard to quarrel with the model from a social perspective.

Somewhere in between are firms (like Rembrandt) that buy into existing disputes and invest substantial resources to develop the dispute. The emphasis is on patent disputes that are already mature, in the sense that the technology already has been deployed in the market, so that the patents write onto existing products. The core competency at which such a firm aims is an ability precisely to estimate the likelihood that the patent will be held valid, the likelihood that the defendant's products will be found to infringe the patent, and the likely amount of damages a court will award for the infringement. Those firms closely resemble the litigation financing firms that have proliferated throughout the legal community in recent years,<sup>97</sup> with the addition of a particular expertise in patent litigation.

(b) Funded by Financiers or Incumbents?

The discussion above also suggests that the identity of the investor in the intermediary should have structural significance. Most of the patent exploitation intermediaries rely on funds from external investors – pure financiers – with no particular

---

<sup>97</sup> E.g., Alison Frankel, *Helping Underfunded Plaintiffs Lawyers – at a Price*, American Lawyer, Feb. 13, 2006.

role in the industry, and no strategic goal other than a substantial return on their investment. Indeed, some intermediaries explain that the detached view of the financier is central to the success of their model because it allows them to pursue patent enforcement and litigation in a clinical and perfectly rational manner, unencumbered by the distractions of social or business relations that would come with funds from an incumbent. This unclouded and calculating rationality that independence seems to be an important cultural attribute of these firms – it came up in one way or another in all of our conversations with representatives of these firms.

But not all firms have that perspective. In particular, Nathan Myhrvold's Intellectual Ventures entities rely on funding from a large set of the most important incumbents in the industry, with the conspicuous exception of IBM. Returning to the discussion above, it is easy to see what Intellectual Ventures provides that the financier-funded intermediaries cannot – the ability to use relational ties to overcome the information paradox that makes it so difficult for independent inventors to obtain fair value for their technology from large incumbent firms. Thus, it should be no surprise to see that Intellectual Ventures appears to have been much more successful than any of the other prominent intermediaries at negotiating license agreements with major incumbent firms.<sup>98</sup>

Looking at the “supply” side of the equation as well, it is easy to understand why incumbents would so willingly invest in an intermediary controlled by a person whom

---

<sup>98</sup> This discussion substantially oversimplifies the business model of Intellectual Ventures. A large part of the acquisition strategy of Intellectual Ventures is wholly unrelated to any reasonably foreseeable enforcement of the patents. Rather, it is designed to provide freedom to innovate in areas in which Intellectual Ventures hopes to be filing its own patent applications based on its own innovative activities.

they know (like Myhrvold). If the incumbent firms believe that they are wasting resources on litigation against “trolls” because of legal rules that (in the view of incumbents) give trolls an unfair return on their patents, the natural response of a rational incumbent would be to invest in the “troll” directly, so that the incumbent could recover the “unfair” returns that the troll earns in litigation against the incumbent. Whether incumbents are correct in that assessment of the situation, it seems fairly clear that it is the view of many of the incumbents and that something like that view has helped to drive the investments in the Myhrvold enterprises.

In any event, the investments should be privately profitable if Myhrvold can obtain license revenues from the incumbent firms that are adequate in light of the funds he spends to acquire patents. From a social perspective, assessing the value of the activity is similar to the assessment of the firms discussed above. At bottom, the question is whether the incentives arising from the funds that flow to original inventors through Myhrvold’s patent acquisitions exceed the drag on innovation reflected by the licensing fees that the incumbent firms pay. Given the relatively low transaction costs of the licensing, it is difficult to quarrel with the model from a social perspective (at least in the absence of some problem with the quality of the patents that Myhrvold acquires and enforces).

## **V. CONCLUSION**

If there is a puzzle that motivates this paper, it is the question why the pharmaceutical and manufacturing industries can have monolithic perspectives on patent policy while each firm in the software industry seems to have a different position on patent policy, a position which is likely to change from time to time. The usual answer is



that the low ability of patents to appropriate innovation in the industry makes the patents less useful and thus lowers the social value of granting them. But that explanation would support a pattern of persistent ambivalence. What we see on the contrary is heated disagreement and instability of perspective.

We think the history and data presented here provide a useful lens for understanding the views on patent policy of those in the industry. For one thing, the role of patents has changed dramatically from the mid 1960s to the present. In the mid 1960s, patents only had value for firms hoping to challenge IBM's dominance in the production of computers and software. By the 21<sup>st</sup> Century, in contrast, the industry has matured into a complex pattern, with at least four distinct groups (incumbents, venture-backed firms, open source communities, and independent inventors and associated intermediaries), each with a different relation to the patent system.

For many years, there was a debate over the fundamental question of patentability, a debate which remains open in the EU to this day. In recent years, however, policy debates have shifted away from that question to converge on the idea that the system is broken in ways that call for changes in various details of patent policy. Thus, the main area in which there is a realistic likelihood of retrenchment on patentability in this country is in the area of business methods. Because of the common confusion of software patents with business method patents, this presents a potential problem for software firms. The natural response is to concede the propriety of cutting back on the permissibility of business method patents, by emphasizing the distinction between those (bad) patents and the (good) technological method patents on which large software firms like IBM rely. IBM argued in *Metabolite*, for example, in favor of

ratcheting up the “useful application” standard to require some “technical application,” which might bar the issuance of many business method patents.<sup>99</sup> If many business methods are likely to earn a return in the market without the additional power conferred by a patent grant, then such a limitation might be positive.

Another important area likely to see reform is the topic of patent quality. Much of the criticism of patents focuses on a small number of highly visible and dubious patents. Academics for several years now have been raising concerns about the quality of PTO patent review.<sup>100</sup> Those concerns focus on the difficulties that the PTO has faced in identifying the relevant prior art, especially in new technologies like software. The obvious policy response, if the details can be worked out, is some form of “community patent review,” in which firms in the area of a proposed patent would have an opportunity to suggest relevant prior art before a patent is issued.<sup>101</sup> Similarly, on the PTO’s side, the persistent public complaints about notoriously bad patents have resulted in a pointed emphasis on patent quality in the USPTO’s proposed strategic plan for 2007-12.<sup>102</sup> The central concern is that without serious review of patent applications it makes little sense

---

<sup>99</sup> See Brief of International Business Machines Corporation in Support of Neither Party, *Laboratory Corp. of America Holdings v. Metabolite Laboratories, Inc.* (No. 04-607) (O.T. 2005). The patents at issue in *International Business Machines Corp. v. Amazon.com, Inc.*, Nos. 9:06CV242 & 606CV452 (E.D. Tex.) appear to be precisely the kind of “technical method” patents that would be upheld under the position IBM urged in *Metabolite*.

<sup>100</sup> E.g., John R. Allison & Mark A. Lemley, Empirical Evidence on the Validity of Litigated Patents, 26 AIPLA Q.J. 185, 205-07 (1998).; Bhaven N. Sampat, Determinants of Patent Quality: An Empirical Analysis (Sept. 2005 working paper) (patent examiners are less likely to discover non-patent prior art and foreign patents).

<sup>101</sup> For details on the progress of that initiative, visit [www.communitypatent.org](http://www.communitypatent.org).

<sup>102</sup> Available at <http://www.uspto.gov/web/offices/com/strat2007/>.

to continue to give the patents the presumption of validity that traditionally goes with issuance.

Even if those criticisms are anecdotal, most of the large patenting firms agree that the PTO could do a better job of locating relevant prior art and processing applications expeditiously. Thus, although those firms rely heavily on patents, it makes sense for them to support changes that would “raise the bar” of patentability. The support for “gold plated patents” suggests that those firms, for the most part, believe they would do just as well under a system in which it was harder to obtain patents than it is now. Similarly, the positions taken in the *KSR* case suggest a general consensus, at least among large firms, that the standard for obviousness should be changed to make it easier for the PTO to reject patents.

Because these kinds of reforms affect only firms that apply for patents (or who compete in sectors with those who do), debate over them has been relatively technical. That is not to say that the reforms affect all firms equally. For example, proposals for community patent review promise more benefits to large firms with substantial patenting infrastructures than they do to smaller venture-backed startups. It is to say, however, that the various initiatives have gotten more attention from the large incumbents than from smaller firms.

The most prominent topic for debate has been the need for litigation reform. Limitations on the availability of injunctive relief have appeared in recent patent reform bills and were urged upon the Supreme Court in *eBay v. MercExchange, LLP*.<sup>103</sup> Compared to patent quality issues, rules on patent enforceability will affect different

---

<sup>103</sup> 126 S. Ct. 1837 (2006) (No. 05-130).

strategies in markedly different ways. So, for example, in *eBay* representatives of the independent inventor community and other technology licensors like Qualcomm predictably pressed for all but automatic injunctive relief.<sup>104</sup> By contrast, representatives of the open source community and firms that are more likely to be defendants in patent litigation lined up to argue vigorously for limitations on injunctive relief. Indeed, many of them argued for a categorical bar on injunctive relief in favor of “non practicing entities,” a position received sympathetically by some justices.<sup>105</sup>

Many industries consolidate as they mature into a small group of relatively homogeneous firms. If that ever happens in the software industry, it will not happen soon. Business models in the software industry differ starkly from firm to firm. The still-increasing variety in the uses to which software is put offers one reason. Another comes from the common tactic of technology companies to leverage competencies at one part of the value chain against commoditization at another – IBM supporting “free” software that is compatible with its proprietary hardware, software, and service offerings, Adobe supporting one free product that facilitates use of its higher-end proprietary products, even Microsoft supporting free small-business accounting software that should increase demand for its proprietary product line.<sup>106</sup> The variety of strategies, coupled with the relative difficulty of using IP to appropriate innovation in software, underscores

---

<sup>104</sup> See briefs filed in support of respondent by Keara A. Bergin et al.; Martin Cooper et al.; Qualcomm Incorporated and Tessera, Inc.; Wisconsin Alumni Research Foundation et al.; Rembrandt IP Management, LLC; United Inventors Association and Technology Licensing Corporation; Pharmaceutical Research and Manufacturers of America.

<sup>105</sup> See briefs filed by Business Software Alliance and Research in Motion, Ltd.

<sup>106</sup> Candace Lombardi, Microsoft Tries to Lure “Mom and Pop” Companies, CNET, Oct. 30, 2006.

the importance of attention to context in designing IP rules for the industry. The fact that every sector is offering views that support its own interests does not mean that policymakers should ignore those views; it means that they should be sure that any reforms they adopt don't accidentally elevate the temporary interests of firms using one strategy over those of firms using another.