July 27, 2004 Draft

DO PATENTS FACILITATE FINANCING IN THE SOFTWARE INDUSTRY?

RONALD J. MANN*

Table of Contents

---

* Ben H. & Kitty King Powell Chair in Business and Commercial Law, Co-Director, Center for Law, Business & Economics, University of Texas School of Law. I thank Allison Mann for sage counsel of all kinds and Abner Greene for diligent research. For comments, I thank Oren Bracha, Sam Dinkin, Doug Lichtman, John Duffy, Dennis Karjala, Mark Lemley, Josh Lerner, Rob Merges, and Jay Westbrook. I also thank participants at the 2004 annual meeting of the American Law and Economics Association, at a faculty colloquium at the University of Texas School of Law, and at the Symposium on Entrepreneurship and Innovation at the Berkeley Center for Law and Technology.

I am especially grateful to the following businesses (and their respective officers) for their cooperation in obtaining information for this project: 360 Commerce (Jerry Rightmer), Alkera, Inc. (Sam Dinkin), Andrews & Kurth, LLP (Ed Cavazos, Kin Gill), Austin Ventures (Rob Adams), Bluecurrent, Inc. (Shawn Thomas, CEO), Brown University Research Foundation (William Jackson), Buchalter Nemer Fields & Younger (Robert E. Izmirian), Builder Homesite (Tim Costello), Center Point Ventures (Eric Jones), Comerica (Timothy Klitch, Philip Wright), Composite Software (Michael Abbott), Cooley Godward LLP (Pamela J. Martinson), Crave Technologies, Inc. (York Eggleston), Datacert (Alan Harding), DNA Software, Inc. (Mark A. Kielb), Forgent (Ken Kalinoski), Gray Cary Ware & Freidenrich LLP (Craig M. Tighe), IBM (Steve Myers, Manny Schecter), Journee (Rob Beauchamp), Kleiner Perkins Caufield & Byers (John Denniston), Launch Pad (Hambleton Lord), Lombardi Software (Rod Favaron, Phil Gilbert), Microsoft (Tim Cranton, Dan Crouse, David Kaefer, Anne Kelley, Jeffrey M. Koontz, Annmarie Levins, Thomas C. Rubin, Cory Van Arsdale), Murphree Venture Partners (Dennis E. Murphree, Tom Stephenson), Nanostream (Steve O'Connor), New Enterprise Associates (Jimmy Treybig), Palomar Ventures (Jim Gauer), Pervasive Software (David Sikora), Sentient Ventures (David M. Lee), Silicon Valley Bank (Marc Cadieux, Andy Enroth, David A. Jones, Derek Witte), Storm Ventures (Sanjay Subhedar), University of Texas (Bobby R. Inman, Steven P. Nichols), Troux Technologies (Hank Weghorst), Vieo (Tom Bishop), Waveset (Michael D'Eath), Wilson Sonsini Goodrich & Rosati (Andrew J. Hirsch, John Mao, Kathleen D. Rothman).

Abstract

This paper is the first part of a wide-ranging study of the role of intellectual property in the software industry. Unlike previous papers, which focus primarily on software patents – which generally are held by firms that are not software firms – this paper provides a thorough and contextually grounded description of the role that patents actually play in the software industry itself

The bulk of the paper considers the pros and cons of patents in the software industry. On the positive side, the paper starts by emphasizing the difficulties that pre-revenue startups face in obtaining any value from patents. Litigation to enforce patents is impractical for those firms. Efforts to obtain patents divert the firm's focus from the central task of designing and deploying a product, and the benefits of excluding competitors are limited for firms that cannot themselves exploit the relevant technology. Once the firm is larger, a number of potential benefits appear. First, despite concerns that patents are not effective to appropriate innovation in the software industry, a substantial number of software startups do have patents of sufficient strength to exclude competitors. Because the principal targets of those patents are much larger firms, that finding is important because it suggests that patents are more beneficial to small firms than to large firms. The paper then considers indirect effects related to the use of patents in cross-licensing transactions and in providing information about the firm. The first benefit may be substantial to firms that obtain patents, but the paper dismisses use in cross licensing as a net benefit to the industry: absent some other benefit, all firms would be better off saving the costs of obtaining patents. The information benefits, in contrast, seem to be net improvements in the system of innovation. The question, however, is whether those benefits are sufficiently substantial to justify the costs of obtaining the patents.

The paper then turns to the prominent claims advanced by Larry Lessig, Jim Bessen, and others that the enforcement of software patents has hindered innovation in the software industry through creation of a patent "thicket." The paper rejects those claims for two broad reasons. First, notwithstanding the empirical analysis of R&D spending in papers by Bessen, Maskin, and Hunt, I argue that direct evidence of high R&D spending in the software industry undermines claims that software patents *cause* firms to reduce R&D spending. Second, I argue that the actual structure and practices of the industry belie any claim of a patent thicket. Relying on interviews that I conducted and publicly available information, I show that young firms in the software industry are not in any significant way constrained in their development activities by the existence of large patent portfolios in the hands of incumbent firms.

The paper also contextualizes the role of patents by examining the relatively weak protections that copyright and trade secret can afford. At bottom, neither of those systems can provide a useful mechanism that would allow small firms to appropriate the values of their inventions. If such protection is a significant positive benefit of the patent system, it is equally true that neither copyrights nor trade secrets are (or can) contribute significantly in that respect, however useful they might be in other roles (such as preventing piracy).

The paper closes by considering critically the possibility of middle-ground responses that would limit patent rights in the industry but not abolish them entirely. First, I criticize a possible registration system that might provide the information benefits discussed in Part III without the costs of excluding competitors. I argue that such an approach would be impractical both because it would be difficult to disentangle the information benefits from the right to control technology, and because of my sense that software firms would have an inadequate incentive to participate in such a system. Finally, I consider the possibility of special limits on the rights of "trolls," small non-operating firms formed solely to litigate patents. I argue that trolls serve a useful function as specialized intermediaries and that in fact they may have a positive role in promoting innovation in the industry.

DO PATENTS FACILITATE FINANCING IN THE SOFTWARE INDUSTRY?

I. Introduction

The U.S. software industry is characterized by astonishing levels of growth, innovative activity and competition. Some argue that innovation in the software and related industries has driven much of the innovation in other industries in recent decades.[1] Federal government statistics suggest it is one of the few information technology sectors that consistently showsa large trade surplus,[2] and as the pressures of globalization dilute the comparative advantage of American employees in many sectors, it is worth noting the remarkable level of employment growth over the last decade, from 854,000 jobs in 1992 to more than 2.1 million jobs in 2000 (a 12 percent annual growth rate).[3]

Academics, however, generally see a different picture. They see an industry burdened by an intellectual property (IP) system that grants so many software patents[4] that small companies cannot effectively innovate.[5] That perspective interested me for several reasons. First, unless it is merely a broader attack on the entire IP system,[6] it assumes that innovation in software is so different from innovation in other areas that traditional IP protections are inappropriate. It also jarred with my general skepticism about the deterministic effect of legal institutions. My

---

[1] MOWERY & NELSON 1999.

[2] *See* UNITED STATES DEPARTMENT OF COMMERCE, ECONOMICS AND STATISTICS ADMINISTRATION, DIGITAL ECONOMY 2002, p. 53 (noting trade surpluses in the software industry of more than $2.5 billion a year during the late 1990's).

[3] UNITED STATES DEPARTMENT OF COMMERCE, ECONOMICS AND STATISTICS ADMINISTRATION, DIGITAL ECONOMY 2002, p. 43. During that period, the wages earned by employees grew at an average annual rate of 7.8 percent, for an average wage in 2000 of $80,900, the highest in any of the information technology-producing industries

[4] As I explain below (text accompanying note 8), it is difficult to get precise numbers. It is clear, however, that the PTO is granting far more than 10,000 software patents each year. Allison & Lemley 2000 finds 18,000 software patents during a two-year period from 1996-1998. Their number is extrapolated from a sample of all patents issued during a two-year period, using a methodology that treats a patent as a software patent only if it is "completely embodied" in software. Allison & Lemley 2000:2110, 2115. Greg Aharonian's somewhat broader measure (which appears to include any patent that includes an element of software) estimates 13,000, 17,500, and 22,500 in 1997, 1998, and 1999 respectively.

[5] *See* LESSIG 2001:205-15; Bessen & Maskin 2002; Bessen 2003; Shapiro 2001; FTC REPORT:52-53 (reporting testimony at FTC hearings on the role of patents in the software industry).

[6] John Barton has a broader criticism. He argues that the growth of IP lawyers at a faster pace than R&D spending shows a serious problem in the design of our patent system. Barton 2000; *see also* Kash & Kingston 2001 (arguing that patents do not work in complex industries because they are used as bargaining chips). Doubts about whether the patent system as a whole causes an increase in innovation are not new. *See* Plant 1934:33-37.

intuition was to doubt that legal rules granting patent protection could have a sufficiently substantial effect to alter the course of innovation in *either* direction.

The existing literature on the subject[7] focuses on the nature and effects of software patents. That is to say, most writers have proceeded by identifying patents that fall within the PTO or IPC classes that correspond most closely to software innovation and then examining data about the performance and behavior of the firms that hold those patents. That approach – although useful in examining the nature of software patents and the work of the PTO – has two major disadvantages for the broader agenda of evaluating the effects of software patents. First, the quality of the research depends entirely on the propriety of the definition of a "software" patent. Because software is a recently devised technology, however, it does not fall naturally within any particular class or classes. Thus, any definition that relies on patent classes is to some degree arbitrary.[8] Moreover, large manufacturing firms (Ford, GM, etc.) outside the software industry hold the overwhelming majority of the patents that those papers analyze. Thus, it is unclear whether the empirical results reflect the effects of the software patents or whether similar results would be obtained for firms that receive substantial revenues from the sale of software products or services.

This paper rejects that approach, opting instead to analyze innovation in the software industry itself.[9] My approach is to identify firms that develop software and then to study the effects of IP on the behavior of those firms. Thus, I focus on firms like IBM, Microsoft, and their smaller competitors that often are ignored in the existing research. Another aspect of my project is that it examines the smaller firms in the industry, rather than looking exclusively at large publicly traded companies.[10] Given the importance of small companies to software innovation (a major theme of this paper), that extension is also a substantial advance.

My methodology is empirical and analytical. I rely here on a set of about 60 interviews[11] with a variety[12] of professionals knowledgeable about the software industry: software

---

[7] Although Lessig publicizes the issue, he does not present any new data. His contribution is to provide a perspective on the implications of the existing literature. LESSIG 2001. The most important empirical contributions to the debate are Graham & Mowery 2002 and a series of unpublished papers by James Bessen and his co-authors: Bessen & Maskin 2002, Bessen 2003, Bessen & Hunt 2003. The description in the text refers to those papers collectively.

[8] *See* Allison & Lemley 2000:2115 & n.51: Bessen & Hunt 2003:8-9.

[9] In this respect, my work is parallel to the work of Hall & Ziedonis on the semi-conductor industry. Hall & Ziedonis 2001.

[10] Graham & Mowery 2002 also analyzes patents in the software industry. That paper makes many contributions, but it necessarily focuses on publicly traded firms for which quantitative data is readily available. Another useful analysis of the role of innovation in the industry is Merges 1996a, which compares innovation in the United States and Japan. However, that paper was written before the rise in patenting that has sparked the present debate.

[11] A methodological appendix summarizes the protocols that governed the interviews.

[12] I spoke to twenty executives at startup firms, thirteen investors, thirteen executives at large firms, six executives at banks, and six lawyers.

developers, venture capitalists, angel investors, banks that lend to software startups, large software and hardware firms, and knowledgeable attorneys. Those interviews are designed to provide qualitative information about the motivations and practices that form the institutional environment within which software firms operate.

Analytically, I connect the interviews to several well-developed literatures relevant to the questions that the project raises. Initially, I account for a substantial body of doctrinal scholarship examining the question of how to accommodate existing bodies of IP law to the nature of innovation in the software industry.[13] Because my goal is to understand the relation between IP and innovation, I also must engage a rich and varied economic analysis of innovation. That literature includes formal and informal analyses of how best to allocate the profits of ideas between the various actors in a sequential scheme of innovation,[14] historical analyses about the effects patents have had over time,[15] and empirical analyses (primarily questionnaires) regarding the value of patents in appropriating the profits of innovation in various industries.[16] Finally, because I am interested in the ability of patents to facilitate financing of software firms, I examine empirical studies of entrepreneurial innovation, which consider the nature and effects of venture capital investing.[17]

It should be clear that any effort to examine the relation between patents and innovation must proceed with modest goals. It is not plausible to think that researchers could obtain the evidence necessary to determine whether patents cause innovation in the industry to proceed with an optimal rate in optimal directions.[18] Thus, my work here is consciously imprecise. My goal is to provide a richer understanding of the possible effects that patents have in the industry. Using this methodology, I can only exclude explanations that are inconsistent with events "on the ground." I cannot hope to provide a comprehensive or definitive account of the effects of patents on innovation. With that in mind, I set the stage in Part II with an overview of the structure of the software industry and the debate about patenting in that industry. I follow with

---

[13] Mark Lemley has argued that several characteristics of the industry justify giving software patents a narrow scope. Burk & Lemley 2003:82-90; Cohen & Lemley 2001. More broadly, the early work of Samuelson, Davis, Kapor & Reichman 1994; *see* Samuelson 1990, advocates a *sui generic* scheme designed especially for the industry. At this time, such a scheme probably would conflict with our obligations under TRIPS. *See* Burk & Lemley 2003:110-11.

[14] The work of Suzanne Scotchmer is impressive. *See* Scotchmer 1991; Green & Scotchmer 1995; Samuelson & Scotchmer 2002. For a useful survey of the literature, see Mazzoleni & Nelson 1998.

[15] Merges & Nelson 1990; Merges & Nelson 1994; Moser 2003. Merges also has written about the likelihood that businesses will develop private institutions for disseminating technology. Merges 1996; Merges 2001; Merges 2004.

[16] Levin, Klevorick, Nelson & Winter 1987; Cohen, Nelson & Walsh 2000.

[17] *See, e.g.*, GOMPERS & LERNER 2000; GOMPERS & LERNER 2001.

[18] Most modern studies assume that increases in innovation are uniformly good and thus do not consider the possibility that the patent system might cause excessive innovation. *E.g.*, Merges & Nelson 1990:878 (recognizing the problem, but explicitly assuming that more innovation is better). The classic counterexample is Barzel (1968) (formal analysis of the possibility that patents will cause innovation that is greater or earlier than optimal).

three substantive Parts that discuss in turn the potential benefits of patents in the industry, the potential costs of patents in the industry, and the role of copyright, trade secret, and other alternative schemes that a firm might use to protect its software-related innovations. I close with a brief discussion of problems with possible alternative systems for protecting innovation in the industry.

II. The Software Industry

It is important to begin with an understanding of the structure of the industry. The industry is young. It generally is regarded as originating in the mid-1960's.[19] The concept of the software product – designed by one firm and sold to a second firm for use on that firm's computer – first originated because of the increasing complexity of software[20] and the shortage of the labor needed for each firm to make its own software.[21] The most crucial event was the decision of IBM in late 1968 to "unbundle" its software from its hardware.[22] Sales of software products grew rapidly throughout the 1970's. By the 1980's, the United States had a large and well-developed corporate software products industry, with more than 1800 firms.[23]

The industry was not, however, fated to retain the unitary status that it had when it first evolved out of the IBM-dominated days of the 1960's. On the contrary, the last quarter-century has seen a succession of events that have repeatedly restructured the industry. The number of developments makes any list of key events arbitrary, but for my purposes, the first salient landmark in the fractionation of the industry was the introduction of the personal computer in the mid-1970's. That development rapidly led to a largely separate set of companies producing software for personal computers.[24] The popularization of the graphical user interface in the early 1990's brought with it an increasingly large role for Microsoft, but to this day dozens of competitors continue to provide significant products for those machines.[25] Yet another sector of the industry that arose by that time is the massive sector producing games and other entertainment software.[26] Finally, the rise of the Internet has brought first a tremendous influx of

---

[19] *See* CAMPBELL-KELLY 2003:89-119.

[20] Observers at the time – including IBM management – were profoundly shocked when it took 5,000 man-years for IBM to develop its OS/360 program. CAMPBELL-KELLY 2003:95.

[21] The shortage was driven in part by the rapid deployment of general-purpose computers: the number in the U.S. grew from 4,400 in 1960 to 48,500 in 1970. CAMPBELL-KELLY 2003:90; RUTTAN 2001:338.

[22] Although IBM has more complicated explanations for the decision to unbundle, external observers attribute the decision to pressures from antitrust litigation. *See* CAMPBELL-KELLY 2003: 109-10.

[23] *See* CAMPBELL-KELLY 2003:121-73.

[24] *See* CAMPBELL-KELLY 2003:201-28; RUTTAN 2001:338-39; Graham & Mowery 2002:3-4.

[25] *See* CAMPBELL-KELLY 2003:231-66.

[26] *See* CAMPBELL-KELLY 2003:269-301.

capital into the industry and then a subsequent crash and weeding out when companies were not able to produce results justifying the elevated equity valuations of 2000 and 2001.

As I write, a major fissure is developing between proprietary and open source models of software development. The traditional model of development has been a proprietary one, where a firm develops a product and then profits by sales of that product. Recently, some firms have rejectedthat model , at least in part, to engage in open source development. Open source development generally proceeds on the premise that software products developed under that model will not be subject to the proprietary control of any individual or firm.[27] I discuss the relation between open source development and patent protection in subpart IV(C), but two points are important to the present discussion. First, the demarcation between the two models is not as complete as open source proponents might suggest. Rather, proprietary firms take advantage of open source development under many business models.[28] Second, to the extent there is a demarcation, this project is focused on firms that are expected to receive revenues from the sale of software products.

A remarkable feature of the industry as it has matured is the lack of concentration – a facet that has considerable implications for the competitive structure of the industry and its openness to innovation.[29] Although press reports (and much of the academic writing as well) are preoccupied with concerns about the dominance of Microsoft, the industry is populated with an unusually large number of significant commercial players. Census Bureau statistics report more than 40,000 firms in the industry as of 2000.[30] Nearly 500 firms in the industry had a million or more dollars in sales in 2003, even after the contractions in the industry at the turn of the millennium.[31] In 2002, 209 firms received their first round of venture capital financing, a total of $872 million (an average of more than $4 million for each firm) during a markedly down year.[32]

---

[27] For a brief overview, see Zittrain 2004.

[28] A common tactic is to rely on open source development and consequent standardization of auxiliary products in which a firm is unlikely to obtain a comparative advantage, coupled with profitable marketing of an ancillary product or service over which the firm can maintain such an advantage. *See* FINK 2002 (discussing how proprietary firms can profit from association with open source products); LEONARD 1995:18-27 (discussion of core capabilities and their importance to strategic decision making); *see also* MANN & WINN 2005 (discussing hybrid open source licenses that permit proprietary exploitation of derivative products).

[29] The fractionation is not new, *see*, *e.g.*, CAMPBELL-KELLY 2003:167 (noting that the software industry by the early 1980's was much less concentrated than the parallel hardware industry out of which it had grown), but it has accelerated since the rise of the Internet.

[30] I aggregate data from NAICS 5112 (Software Publishers) and 541511 (Custom Computer Programming Services. The data are available at http://www.census.gov/epcd/ec97/industry/E541511.HTM and http://www.census.gov/epcd/ec97/industry/E5112.HTM (both last visited on Oct. 7, 2003).

[31] The information is available at www.softwaremag.com (last visited May 6, 2004).

[32] During 2002, 652 software companies received a total of $4.3 billion (that is, 443 firms received second or subsequent rounds during 2002). Since 1995, 2907 new firms have received venture capital financing. 2003 NATIONAL VENTURE CAPITAL ASSOCIATION YEARBOOK 40.

Moreover, despite the existence of some prominent firms, the number of large firms is very small – there are only three software firms in the current Fortune 500. Indeed, the top ten firms in revenue had less than thirty percent of the revenues of the industry as a whole.

Another important point is the ebb and flow of IP protection for software for much of the industry's history. Although the form of the protection has changed over time, I share Rob Merges's view that "[t]he United States has traditionally embraced strong protection for computer software."[33] In the early days, it was generally believed that it was "trivially easy to replicate" the software program of a competitor.[34] When initial efforts by major industry players to obtain patents on their products were unsuccessful,[35] firms (and Congress) turned to copyright as an alternative.[36] The Copyright Office formally decided to permit registration of programs in the mid-1960's.[37] Initially, this was a promising arrangement, based on an analogy of literary expression to the lines of code of which a software program is composed.[38] Thus, until the late 1980's, copyright provided relatively strong protection for software.[39]

Over the years, however, as the courts became familiar with software cases, the courts narrowed copyright protection so that it ceased to provide robust protection.[40] The problem that courts increasingly confronted was that "there is nothing in the statute nor in the legislative history to indicate that Congress intended for copyright to protect the *results* (that is, behavior) brought about by the execution of program instructions."[41] Thus, in *Computer Associates v. Altai* in 1992, the Second Circuit adopted a "hard-look" framework that made it difficult to obtain copyright protection for the broader structural features of programs. The court limited protection to specific pieces of the program limited to "expression."[42] Two years later, the Ninth Circuit refused to protect Apple's graphical user interface from appropriation by Microsoft.[43] The façade of pervasive copyright protection came crashing to a definitive ruin with the

---

[33] Merges 1996a:277.

[34] CAMPBELL-KELLY 2003:107.

[35] *See* CAMPBELL-KELLY 2003:107; Menell 2002:15-16 (discussing *Gottschalk v. Benson*, 409 U.S. 63 (1972)).

[36] Congress codified a definition of computer program as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101. *See* Menell 2002:16-20 (discussing that history).

[37] Menell 2002:16.

[38] *See, e.g.*, Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3rd Cir. 1983) (copyright protection for Apple's code).

[39] Merges 1996a:277-78; Menell 2002:19-20; Graham & Mowery 2002:7-8.

[40] Graham & Mowery 2002:8-9.

[41] Samuelson, Davis, Kapor & Reichman 1994:2351.

[42] 23 U.S.P.Q.2d 1241; Merges 1996a:277-278.

[43] *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.2d 1435 (1994); Menell 2002:45-46.

celebrated decision of the First Circuit three years later in *Lotus Development Corp v. Borland Int'l, Inc.*[44]

Yet, long before those decisions limited the overarching importance of copyright protection, major firms in the industry had begun to turn to patent protection. Direct protection of software patents was difficult in the wake of the Supreme Court's 1972 decision in *Gottschalk v. Benson.*[45] Still, several of my interviews suggest that software patents were easy to obtain. Because much of the software through the 1980's was produced by hardware firms, patents easily could be obtained on an object (a microprocessor), programmed to accomplish the relevant function.[46] To be sure, that artifice was not effective in the 1990's when pure software firms like Microsoft started to play major roles, because those firms could appropriate the innovation of those patents in a software program without infringement. By that time, however, patent doctrine had changed so that patent protection was available, at least to those firms that were sufficiently familiar with the process to exploit it.[47]

In sum, despite the contrary mythology of a golden age of IP freedom, it is not clear that there was any time when software was not protected by IP. When copyright protection seemed likely to provide adequate protection, many who were active in the industry thought that patent protection would be counterproductive.[48] Nevertheless, as it became increasingly clear that copyright protection would be inadequate, the supporters of patent protection in the industry gained force, so that many of the leading firms now have large numbers of patents. This Article considers the role those patents have come to play as the industry has matured.

III. Do Patents Induce Commercialization in the Software Industry?

As suggested above, it is difficult to develop any concrete understanding about the effects patents might have on innovation in a particular industry. Even if we assume that all increases in innovative activity are positive,[49] it is difficult to separate economic effects related to the legal artifact of patent protection from the effects of the innovation for which the patent is granted.[50] Recognizing that difficulty, this Part of the Article examines the potential positive effects of patents in the software industry.[51]

---

[44] 49 F.3d 807 (1st Cir. 1995) (no copyright protection for pull-down menus in spreadsheet program).

[45] 409 U.S. 63 (1972) (barring patent on algorithm); *see* Menell 2002:16.

[46] Witek Interview; Hill Interview

[47] Merges 1996a:278-81; Menell 2002:15; Graham & Mowery 2002:9.

[48] LESSIG 2001:207-08; Menell 2002:15; Graham & Mowery 2002:10-11.

[49] *See supra* note 18.

[50] *See* Moser 2003.

[51] The closest approach to such an examination in the existing literature is the discussion in Merges 1996a that suggests that the stronger patent protection in the United States (as compared to Japan) helped support the development in the United States in the early 1990's of custom software developers.

To understand the effect of patents on the development of software, I focus on small firms, which typically are venture backed. Several related considerations support that choice. First, and most obviously, many of the most important innovations in the software industry come from relatively small firms. Yet the scholarship to date focuses exclusively on large publicly traded firms. Second, the complex capital arrangements of public firms make it harder to analyze the relationship between patent portfolios and the flow of capital into and out of those firms. Thus, a study of the simpler arrangements of venture-backed firms is preferable. The final point relates to the nature of qualitative interviews. Generally, it is easier to obtain reliable interviews from smaller firms and their venture capital investors than from large firms. It is less common at a large firm to find a person with complete hands-on responsibility both for the financing arrangements and for the policies with respect to the development and protection of IP. Executives at larger firms also are much more likely to articulate views constrained by the legal positions underlying the broader interests of the firm, and thus relatively unlikely to engage in the kinds of wide-ranging conversations likely to provide useful information to the quasi-anthropological research that I conduct. Moreover, the venture capital investors have highly diversified experiences relevant to my inquiries. Many have experienced good and bad returns on literally dozens of investments, with considerable insight about what makes the investments good and bad, and many have had previous careers as entrepreneurs themselves, giving them a more complete perspective.

This Part proceeds in three steps. First, I discuss the goals of the venture capital investors, which relate to the likelihood that a portfolio firm can differentiate itself from its competitors in a reliable way. Second, I analyze the ability of patents to satisfy that goal. Although the discussion evinces pessimism about patents as a mechanism for appropriability in the software industry, it also emphasizes a shift in the efficacy of patents as firms grow. Thus, I discuss a number of reasons why patents are not likely to further that goal in the earliest stages of a software firm, before it has revenues or begins shipping a product. Then, I show how once the firm moves beyond infancy – to a stage with revenues or a product – patents can have a variety of beneficial effects. That section provides a framework for relating the evidence drawn from my interviews with the existing analytical literature, teasing out of that pattern a set of direct and indirect positive effects that patents can have in various circumstances.

A. Venture Capitalists and Sustainable Differentiation

The development of software is expensive and time consuming. Thus, it is not common for a successful product to be developed by an individual developer working in his spare time. Rather, most commercial software products are the result of years of effort. That effort, in turn, inevitably requires the expenditure of considerable monetary resources. Of course, young firms can – and normally are expected to – go a considerable way toward developing their concept without using the funds of third parties. At some point, however, they will exhaust their own resources and the readily available resources of friends and family members.[52]

[52] *See* MAY & SIMMONS 2001:32-33.

At that point, in most cases, the firm will turn to institutional investment. One of the most prominent and common[53] sources of that investment is a venture capitalist.[54] Venture capital firms are intermediaries that raise funds from institutional investors (corporate pension plans and the like) and invest those funds in startup companies in technology areas.[55] There is a vast literature on the structure of the industry, including detailed studies of many aspects of the contract structures that those firms use in dealing with their investors and with the portfolio companies in which they invest.[56] For the purpose of this study, however, the structure of the venture capitalist is relatively unimportant. What is important for this study is understanding as precisely as possible what characteristics of a portfolio firm are important in leading a venture capitalist to invest. Although there is little quantitative empirical work on that question,[57] the most obvious role that IP protection might play in that process is that the monopoly that it grants on the exploitation of a covered technology will cause investment to flow into the firm that has created the technology. The monopoly could support such a flow of investment – at least in theory – by creating market power that would allow the firm to earn supranormal profits by exploiting the technology in question.

The first point to understand about startup companies is that the uniqueness of the firm's product is not likely to be one of the primary issues a potential venture capital investor will analyze in deciding whether to invest in the firm.[58] Rather, the investor is likely to start by focusing on issues that validate the core competency of the firm to execute its concept

---

[53] Angel investors provide a source of financing that is parallel (or, in some cases, preliminary) to venture capital investment. Their role in very early stages is evident in the results of one survey finding that angels provide initial funding for more small, high-growth companies than venture capitalists. *See* MAY & SIMMONS 2001:32. That does not suggest that angels finance more firms than venture capitalists, only that they are particularly common in the earliest stages of a firm's development. Even in those cases, venture capital funding is likely to be important at some stage of the firm's development. Angel investors tend to be successful business executives investing the profits from their earlier endeavors in a much less formal way than venture capitalists. The limited interviews with angel investors and the scant literature on the subject suggest that the perspective of the typical angel investor is quite similar to the venture capital perspective. *See* MAY & SIMMONS 2001:170-71 (reporting advice from Guy Kawasaki); Jackson Interview; Lord Interview.

[54] For a discussion of the impact of venture capital financing on small firms and the economy as a whole, see GOMPERS & LERNER 2001:41-83.

[55] For a discussion of how venture capital organizations operate, see GOMPERS & LERNER 2001:87-115.

[56] For a collection of quantitative analyses of those topics, see GOMPERS & LERNER 2000. For a good summary, see Klausner & Litvak 2001.

[57] Hellman & Puri 2000 presents data indicating that venture capitalists are more likely to invest in "innovative" first movers than "imitative" second movers. The generality of that argument deprives it of much force on the questions about IP at the heart of my work.

[58] A typical comment is that entrepreneurs are naïve if they think they have discovered a valuable product idea for which there is no competition. ADAMS 2002:20-21. Indeed, the absence of competition generally is regarded as a bad signal, because it suggests that the idea is not worth pursuing. *See* ADAMS 2002:21; MAY & SIMMONS 2001:170.

successfully.[59] For example, investors will be interested in such things as experience in the relevant market and the skills of the management team.[60] One remarked: "Every company of mine that has failed has been mismanagement of executives, not technical failure."[61] Similarly, even before investors consider whether a firm can protect a market leader position, they will want to know whether the product is one that customers need so desperately that the firm can gain a significant place in the market.[62]

Still, for firms that have a credible product idea and the expertise to implement it, venture capitalists plainly accept the idea that their goal is to identify firms that will have sufficient market power to earn extraordinary profits. IP protection is important only indirectly, as a tool that might provide that market power. The key is "sustainable differentiation"[63]: something special about the particular firm that will enable it to do something that its competitors will not be able to do for the immediate future. The interviews also reflect more picturesque terminology – referring to "secret sauce"[64] or "magic dust."[65] But it is clear that the key to a desirable investment opportunity is in the expectation of market power, with all other attributes of the company being indirect predictors of that ultimate goal.[66]

For example, investors commonly referred to lead-time or first-mover advantages.[67] The premise was that a portfolio company that truly was the first to provide a sophisticated and functional response to an important problem could expect to earn a supranormal return for years to come.[68] Interestingly enough, the expectation rested on the perception that a firm could

---

[59] *See* Harding Interview:2 ("There are competitors that probably have equally as good software [as we do, but] they can't do the implementations [for the customers].").

[60] *See* ADAMS 2002:27-39 (discussing "execution intelligence"), 125-52 (discussing the importance of the management team in securing funding); MAY & SIMMONS, *supra* note 52, at 171.

[61] Lee Interview: 2; *see* Subhedar Interview:1 ("The most important thing is execution.").

[62] *See* ADAMS 2002:49-68 (discussing "market validation" and the need to develop a product that responds to customer "pain"); Gauer Interview:2 ("The point is whether there is a pain point in the market to which these people are going to apply a pain reliever."); Kielb Interview:1.

[63] Lee Interview:1.

[64] Lee Interview:1.

[65] *See* Weghorst Interview:3.

[66] *See* Rightmer Interview:1 (comment of developer that IP "is a check-list item on [VC's] list. What they're really looking for is barriers to entry."); Subhedar Interview:1 ("[T]here could be patent protection, but that in and of itself is not what you're looking for. Really what you're looking for is how are you going to sustain your position.").

[67] *See* ADAMS 2002:73-94; D'Eath Interview:6; E. Jones Interview:2-3 (discussing benefits of an "installed base" of users in maintaining recognition as a market leader); Rightmer Interview:1 (explaining that his firm's success hinges on getting customers to "lock-in" to his product because they can afford to change products "only once every ten years"); Sikora Interview:1.

[68] Hellman & Puri (2000) provide empirical evidence that VCs are more likely to fund startups with first-moving "innovative" products rather than second-moving "imitative" products.

maintain a lead on its rivals as long as it kept improving its technology as quickly as its competitors.[69] I rarely if ever heard investors (as opposed to developers) who expected portfolio firms to obtain and keep a strong market position through "lock-in" or "bandwagon" effects.[70]

That is not to say that IP protection is unimportant. It is clear, however, that different investors have different views about it.[71] Some feel that intellectual property always is important, and claim that they never invest without strong patentable technology.[72] Even those investors, however, go on to say that they are not as interested in the IP protection as in technology that is sufficiently cutting-edge to warrant protection.[73] Others, however, particularly those that emphasize early-stage companies, say that IP protection is unimportant for software investments.[74] Still others take a middle position, holding that IP protection matters some, but not all, of the time.[75] Most of those who addressed the subject recognized differing perspectives on the point and argued that those with the other perspectives are misguided.[76] The most likely explanation is that investors are simply implementing different investment models based on their particular expertise.

---

[69] *See* D'Eath Interview:6; Kielb Interview:2; Sikora Interview:1 (arguing that his firm's lead time of 6 months over its competitors is important: "9 women can't make a baby in one month. There are problems that just take a certain amount of time to solve"); Weghorst Interview:3 ("Sustainable differences are typically time and materials put into [the software]."). {The Sikora quote apparently is an allusion to a famous comment by IBM chairman Tom Watson, Jr. related to development of the seminal OS/360 software product: "The bearing of a child takes nine months, no matter how many women are assigned." Quoted in CAMPBELL-KELLY & ASPRAY 1996:199.}

[70] "Lock-in" or "bandwagon" effects occur when the value of a particular technology increases with the number of other users, and have the potential to allow a particular technology to remain dominant even when later, superior technologies appear on the market. For general discussion, see ROHLFS 2001; *see also* LIEBOWITZ & MARGOLIS 1999 (arguing that inefficient lock-in rarely occurs, with numerous examples from the software industry).

[71] *Cf.* GOMPERS & LERNER 2000:47 (discussing a variety of investment perspectives without specifically referring to IP protection).

[72] *See* Inman Interview:1; Lee Interview:1; Jackson Interview:4; Murphree Interview:1.

[73] *See* Lee Interview:1 (suggesting that the "next step" is whether you want to "open your kimono" a little bit by having the technology registered for protection).

[74] *See* Adams Interview:1 ("Quite frankly from an investor's standpoint, IP protection doesn't mean a whole lot."); Denniston Interview:1 ("Software is unique in that we don't look for IP protection."); Gauer Interview:2 ("[P]atentable work tends to correlate with working in new areas and being world-class in those areas but I would never make an investment decision based on whether there is a patent or not or whether I thought a patent application would be successful."); E. Jones Interview:2 (looking for "something unique" rather than something that is patented because patents are not worth the expense in most software investments); Treybig Interview:1 "[P]atents imply a contribution, it helps evaluate the company and what they're creating that's different, that can let them win.").

[75] Stephenson Interview:1.

[76] *Compare* Inman Interview:1 (criticizing investors who "claimed that IP was nonsense"), *with* Gauer Interview:3 (arguing that emphasis on copyright and patent protection illustrates that the Southern California venture-capital industry is "less mature" than the Northern California industry).

B. Patents and Sustainable Differentiation

If understanding what venture capitalists want is the first question, the second question is understanding whether they believe that patents can provide it. On that point, investors and developers discern a balance of interests that shifts as the firm grows, from the earliest stage, where patents are not often helpful, through intermediate stages to the terminal stage of the venture-backed firm (just before an acquisition or IPO) where patents are almost universally viewed as useful.

1. The Basic Problem: Patents and Appropriability

A basic problem with patents for software firms at all stages is the sense that even with a patent it often is difficult for a firm to "appropriate" the value of its invention.[77] Specifically, my interview subjects agreed that competitor s usually could implement most of the aspects of a software product that a patent might protect without infringing the patent.[78] One reason for that problem is the multifarious nature of software innovation, which permits many solutions to any particular problem. Another contributing factor is the poor match between patents and products in the industry: it is difficult to patent an entire product in the software industry, because any particular product is likely to include dozens if not hundreds of separate technological ideas. Thus, it may take a number of novel ideas – and patents – to build a defensible barrier around a product.[79] Another problem is that the technology tends to develop so rapidly that by the time a patent is issued – and the formal right to exclusivity commences – the technology may be obsolete for all but the broadest patents.[80] Litigation at that point will involve efforts by the patent owner to challenge technology of a subsequent generation where application of the patent may be less clear.[81]

---

[77] The literature on appropriability is Teece 1986 and Levin et al. 1987. The insights of those papers are that (1) the ability of businesses to appropriate the value of innovation differs from industry to industry; and (2) the mechanisms that businesses use to appropriate the value of innovation differ in their effectiveness from industry to industry.

[78] *See* Beauchamp Interview:4 (discussing ease of working around software patents); Harding Interview:1 ("There are a lot of ways to work around patents."); E. Jones Interview:2 ("[I]n software it is so easy to change things that it is so easy to do the same function, but do it in a different way."); Van Arsdale Interview:3 ("Most patents you can get around. * * * * There's always a way to do it different. Some times you have to spend as much money as the patent holder spent, but that doesn't mean you can't.").

[79] A biotech startup, by contrast, can build a defensible barrier around its product with one patent or only a few patents on the relevant composition or process. Thus, a biotech startup more readily can use patents to appropriate the value of its invention.

[80] *See* Rightmer Interview:1 ("The technology moves so fast and the Patent Office moves so slow."); Weghorst Interview:4-5 (explaining that the exclusivity period of the patent is "out of sync" with the timing of the value of the innovation).

[81] *See* Cohen & Lemley 2001.

That problem is not universal. As I discuss below, some patentees manage to obtain patents of sufficient breadth that all possible solutions to an important problem would infringe the patent. More broadly, some knowledgeable observers attribute it not to the nature of software technology, but to the infancy of the industry. Because the industry is developing so rapidly, some argue, the nature of technology and even of technological developments is so poorly understood that firms do not understand the value they could appropriate from patents if they pursued them in an informed way.[82] From this perspective, the relatively limited appropriability provided by software patents should be compared not to the relatively high appropriability of hardware patents,[83] but to the even more limited appropriability that software patents provided a decade ago. The expectation, then, is that in a matter of decades software technology will be as effectively subject to patent protection as the related hardware technology is at this time.

For now, despite those qualifications, the relevant point is that for most firms, most of the time, there is little prospect that the patents they obtain will provide market power that they can use to exclude competitors. That point is underscored by the relative infrequency with which venture-backed software startups have patents. For comparative purposes, consider that only about 20% of venture-backed software companies will have a patent within five years of their first financing (with each of those firms holding about two patents), while more than half of biotech startups will have patents by that time (with each of those firms having about seven patents).[84]

2. Patents and Pre-Revenue Startups

In addition to the problem that patents often are not an optimal mechanism for appropriating the value of software innovation, a number of considerations make it particularly difficult for early-stage companies to employ patents effectively. The key points here are the limited efficacy of litigation for those firms, the constraints on resources that make it infeasible to focus on patenting, and the limited value to pre-revenue firms of excluding competitors.

---

[82] That view is stated most clearly in Dinkin Interview. It resonates strongly with SPAR 2001.

[83] Many of my interview subjects expressed the view that software patents generally are not as valuable as hardware patents. *See, e.g.*, Bishop Interview:2 (endorsing the view that software patents generally are easier to work around than hardware patents, particularly when they are method patents: "Software is so malleable that it is easy to exploit the idea."); E. Jones Interview:1 (explaining that patents on software are generally less useful than patents on hardware); Kielb Interview:1 (emphasizing the range in utility of patents among industries, with software generally at the low end); Treybig Interview:2 (suggesting that hardware patents have more potential to exclude competitors than software patents). For a contrasting view, see Dinkin Interview:12-13 (arguing that the difference in valuation of patents relates to the relative youth of the software industry); Treybig Interview:6 (suggesting that when the industry matures software patents and hardware patents will have similar values).

[84] Details of that empirical research are in a forthcoming paper with Tom Sager. The research is based an a dataset of all venture-backed firms in the software and biotech sectors that received their first financing in 1998 or 1999 (about 800 software firms and 170 biotech firms), analyzing that patents that those firms held on December 31, 2003.

(a) The Perils of Small-Firm Litigation

On the first point, even if an early-stage company had a patent, it is unlikely that it would have available resources to commence litigation to enforce the patent against a competitor.[85] That is particularly true when the competitor is a large firm. One problem is the disparity in litigation resources. One investor emphasized the concern that a large defendant would "rain lawyers on your head and tie you up in court for the next ten years."[86] A somewhat different concern about suing a large firm is the likelihood that the large firm might have a patent that the small firm infringes. If so, the lawsuit might simply alert the large firm to the presence of the small firm.[87]

A related concern is that firm culture is degraded when a firm must rely on licensing revenues instead of developing its own product.[88] Interestingly enough, that sentiment was expressed even at the firms that rely heavily on licensing revenues. Those firms emphasized efforts to maintain a product-centered culture emphasizing production of the firm's own products.[89] One explained the cultural risk as follows:

> You don't need sales people; you need attorneys. You don't need solutions architects; you need accountants. So you wind up losing the very people who are, who were, and who continue to be constructive * * * and innovative and help you build things and would give us a continuing competitive advantage.[90]

---

[85] *See* Gill Interview:1 (emphasizing that the cost of enforcement is more of a barrier than the cost of obtaining the patent); Harding Interview:1 ("We just don't have a large enough war chest at this point in our life cycle. Down the road we might be more aggressive once we have enough cash to do it.").

[86] Murphree Interview:1; *see* Abbott Interview:5; Adams Interview:1 ("The ability to defend your patents is only as big as your bank account. And nobody wants to pump money in to do that before you have money to fund that from operating income. * * * * [I]f your only hope to make the company work is to go to court and win nobody is going to invest."); Beauchamp Interview:5 ("[A]s a start-up, it's unlikely that we are going to leverage [our] patents in any kind of lawsuit."); Weghorst Interview:5 (explaining that a patent would have little value for an early-stage startup because competitors would doubt his will and ability to enforce it). This point is not new, of course. Cohen et al. report a similar finding in their cross-industry surveys. Cohen et al. 2000:14-16.

[87] Rightmer Interview:1. Thus, to that executive at least, it surely came as no surprise when IBM responded in the summer of 2003 to SCO's noted lawsuit regarding Linux with a counterclaim alleging that SCO's software infringes a number of IBM's patents. *See*, *e.g.*, Stephen Shankland, *Big Blue Files Counterclaims Against SCO* (Aug. 7, 2003), available at http://zdnet.com.com/2100-1104_2-5060965.html. For a similar perspective, see VON HIPPEL 1988:53 (discussion of responding to a charge of infringement in the semiconductor industry by mailing back to the complainant "'a pound or two' of its possible germane patents").

[88] Van Arsdale Interview:2. In response to a question, he emphasized that IBM is *not* a counter-example to that reasoning, explaining that despite the "huge asset" of IBM's patent portfolio IBM has managed to maintain a culture firmly focused on developing its own competitive products. *Id.*

[89] Kalinoski Interview 2:4; Thomas Interview 2:3.

[90] Thomas Interview:3.

Indeed, at the one firm that had a major licensing program, the entire program was entrusted to third-party professionals, so that it would not interfere with the focus of the on-site software engineers.[91]

(b) Diversion of Focus

Similarly, many investors and developers emphasized that attention to patents can be damaging to a startup, because it has the potential to divert limited time and resources from what is likely to be a highly time-pressured effort to develop a product and convince customers and investors of its worth before the firm runs out of capital resources.[92] One investor explained: "[We] typically find that the companies that focus on just patents don't have the right view of what is important, and they really are therefore not successful in business. And they're usually not around to prosecute their patents."[93] Developers understand the point well. As one said: "Every dollar we spend on [patenting] is a dollar we can't spend on a software engineer."[94] Another, with a patent-leaning background from his days at IBM commented: "Patentability is something we will pursue, but let's get the product out first."[95]

Thus, a young company is presented with a challenging task. If the nature of the firm's innovation is such that IP is *ever* likely to be important, it must spend sufficient resources on the protection and development of intellectual property from the earliest days of the company – as an investment in the possibility that the firm *might* at some point grow to the point where the IP is useful.[96] The firm that fails to protect its IP at the earliest stage is like a desperate ship at sea that empties its water in the hope of evading a faster pursuer: it might survive for the time being, but it may have sown the seeds of its inevitable failure if it survives to a later stage.[97] On the other hand, it must not spend so much that the company fails before it is able to recoup its investment.

---

[91] Thomas Interview:4.

[92] *See* Denniston Interview:1 ("For Series-A firms, there just isn't the budget for patenting."); E. Jones Interview:3 ("It was better to spend the time continuing to advance the technology than it was to push people off to the side and have them focus on creating the patents and work on it."); Kielb Interview:1 (the cost of diverting the "time, attention, energy, and focus" of personnel to a "suboptimal" use is more important than the monetary cost of obtaining a patent); Treybig Interview:6 ("If you're a small company, and unless you have a hell of a patent, it's pretty hard to spend money on patents versus another salesman or something.").

[93] Subhedar Interview:4.

[94] Harding Interview:1; *see* Bishop Interview:2 (former IBM executive now leading startup explaining shift in philosophy among investors so that now "there is a lot more interest in getting the product out than in having patents"); Rightmer Interview:1 (discussing costs of documentation necessary to protect the ability to obtain patent's on the firm's innovation).

[95] Bishop Interview:2.

[96] *See* Gill Interview:1 (explaining that the strategy is to obtain patents early "knowing that you won't enforce them until later").

[97] The metaphor will be plain to readers of Patrick O'Brian.

Firms have developed a number of strategies for dealing with that problem. Some involve using half-measures to protect the IP, such as filing provisional applications,[98] or omitting standard practices related to documentation of the work of engineers.[99] Those practices do not directly abandon the IP, but they may make it more costly and difficult to protect it in the future. The bottom line is that even for companies that have begun to earn to substantial revenues it often does not seem appropriate to devote the resources necessary to ensure that all of the firm's innovations are patented.[100] Others – it must be said that executives with prior experience at large IP-sensitive firms like IBM[101] or Bell Labs populate this category – seem to relish the discipline of making sure that the IP is pinned down no matter how difficult it may seem to find the time and resources to do so.[102] It is plain that the difficulty of this strategic choice, coupled with the difficulty in accurately predicting the future prospects for their products and their IP, is a problem about which startup software executives worry constantly.[103]

Investors, of course, are aware of this problem. Their approach typically does not extend to forcing (or even urging) their portfolio companies to seek patent protection. However, they do go to considerable lengths to evaluate the IP that potential portfolio companies have. In a typical process, the investor will know most of the reputable patent attorneys in the local community. If one of those attorneys had filed a patent, the VC would discuss the patent with that lawyer. If an attorney with whom the VC did not have a preexisting relationship filed the patent application, the VC would have the patent studied by an attorney in whom it had confidence. In context, it was clear that the intent of the examination was not purely technical – is this a patent likely to be granted – but also a broader exercise to understand what type of market power the patent might

---

[98] *See* Beauchamp Interview:3-4; Weghorst Interview:4.

[99] *See* Weghorst Interview:6. The costs of a vigorous pursuit of that process are considerable. One executive with experience at IBM and elsewhere suggested that as a rule of thumb he commits 4-8 engineer hours *per week* for the life of the application, examining, and issuance process. *See* Bishop Interview:2.

[100] Sikora Interview:1 ("Software companies are not 3M. We don't organize our offices to get patents.").

[101] IBM's dominant patent portfolio is not an accident. It plainly has one of the most – if not the most – patent intensive environments among American companies. *See* Dinkin Interview:3-5. IBM has been the leading U.S. patentee every year since 1993, and IBM also appears to be the leading software patentee. Manny Schecter, *IBM's Strategies for the Creation, Protection and Use of Intellectual Property in Software* (Nov. 30, 2001), available at http://emertech.wharton.upenn.edu/ConfRpts_Folder/WhartonKnowledgeAssets_Report.pdf.

[102] *See* Kalinoski Interview:2-4 (former IBM engineer leading startup company; describing comprehensive incentive program designed to emphasize the importance of patenting to engineering team).

[103] *See* Abbott Interview:5 (explaining that the decision has to be made "looking at the amount of time, dollars, and effort required that potentially could pay off huge in a couple of years"); Kalinoski Interview:6 (discussing that choice and emphasizing that patents can be thoroughly protected only through a conscious commitment: "You have to make a conscious decision – either you're gonna do it, or you're not gonna do it."): Thomas Interview:1 ("[I]t's a mindset issue.").

(or might not) provide.[104] In short, standard industry practice now views examination of that market power as central to careful due-diligence practices.[105] That practice at first seems to be in tension with the thesis of this section – that patents have little value for the earliest-stage startups. In fact, however, it leads into the point of the next section, that the firms that survive their earliest days may reap substantial value from patents.

(c) The Limited Benefits of Exclusion

One final element of the patenting calculus for small firms may seem obvious, but is worth noting. Because those firms do not yet have a product, they have no opportunity for revenues. Thus, the benefits they reap from excluding competitors are minimal at best. Only if they survive to a later stage – where they can hope to profit from their own exploitation of that product – will they be able to reap any substantial value from the exclusion of competitors.

3. The Increasing Value of Patents for Later-Stage Startups

When firms mature to the point of having revenues, the systematic difficulties that plague the efforts of pre-revenue startups to obtain and exploit patents dissipate. That does not mean, however, that patents suddenly are the philosopher's stone that will turn their creative endeavors into IPO's. Rather, the underlying problem of appropriability remains to plague most efforts to use patents directly to exclude competitors. Still, my interviews suggest a series of benefits that patents might provide for later-stage software startups. This section describes those benefits, by reference to the bodies of existing literature that have offered them as theoretical possibilities.

(a) Direct Effects: Protecting a Space for Innovation

The most important point concerns the direct ability of the software patent to carve out for the firm a space in which it can innovate without competition. I explain above that my interview subjects often complain about the difficulty in using patents to exclude competitors. Although there is some truth in both of those points, they are overgeneralizations, at least once a firm reaches the stage at which it has designed a product that it can market to customers.

To respond directly to the first point, it is clear that *some* firms in the industry obtain a substantial amount of revenues by licensing the use of their patents to competitors that need to use the patented technology in their own products. Indeed, even in my limited sample three small Austin companies – Applied Science Fiction,[106] Bluecurrent,[107] and Forgent[108] – have

---

[104] *See* Lee Interview:1: Murphree Interview:1 (emphasizing interest in how "defensible" the market position is); Stephenson Interview:2.

[105] *See* Fine & Palmer (in FROM IDEAS TO ASSETS) (discussing modern due-diligence practice for IP assets); Haller et al. (in FROM IDEAS TO ASSETS) (same).

[106] Urdahl Interview. Details regarding the amount of revenues do not appear to be public. For press releases announcing licensing transactions, see, *e.g.*, http://www.asf.com/events/press/092402_Konica_RS1.shtml,

obtained substantial revenues from patent licenses. I do not believe that industry-wide statistics quantify the size of that market, but it plainly is substantial. Those transactions – and others like them – demonstrate that *some* software patents are sufficiently robust to allow their holders to appropriate substantial value from the underlying inventions. Licensing transactions are noteworthy given the difficulties small firms face in enforcing patents against large firms. As discussed above,[109] the small firm with a revenue-producing product must be quite confident in the value of its technology before it wisely can cross swords with a company like IBM.

More generally, it seems clear that the received wisdom that patents are not useful to appropriate software-related inventions is overstated. Two separate points are important. The first is the distinction between the relative rarity of observed offensive use of patents – for out-licensing or litigation – and the use of patents to exclude competitors. The relative rarity of offensive use of patents does not prove that the patents are not sufficiently robust to exclude competitors. As discussed in the previous section, there are many reasons why a firm might want to wait until late in its development to advertise the nature of its technology and its proprietary claims to that technology.

A firm can refrain from offensive use of its patents and still derive important value from the patents as a device to exclude competitors. Contrary to the perception that patents tilt the playing field in favor of large incumbent firms to the disadvantage of small firms,[110] patents in this context afford a unique opportunity to the small startup.[111] It is as if the patent system grants the small firm an automatic stay of competitive activity, to remain in force long enough for the firm to attempt to develop its technology. For large firms, the marginal increase in appropriability that comes from patents may have little benefit: IBM could compete quite successfully against smaller firms even if it did not have patents protecting its product from copying competitors.[112] For the smaller firm, however, the ability to prevent incumbents like

---

http://www.asf.com/events/press/Gretag.shtml, and http://www.asf.com/events/press/DurstSigma.shtml. Applied Science Fiction now has been acquired by Kodak. www.asf.com.

[107] Thomas Interview; http://online.wsj.com/article_email/0,,SB106677936791434300-H9jeoNjlaR2nZyqZnqHcaeHm4,00.html.

[108] Forgent's patent on the .jpeg file format has produced tens of millions of dollars in revenues. http://www.forgent.com/company/press_room/dallas_082802.shtml.

[109] *See supra* note 87.

[110] I discuss that perception in detail in Part IV.

[111] My analysis here resonates with the general discussion in Barnett 2004. If we differ in anything other than my contextual approach, it is in my view that the benefits of patents arise from a much larger number of interrelated effects than those that he discusses, and in my view that patents are quite valuable for large firms in addition to small firms.

[112] That is not to say that patents are not valuable to IBM. As discussed below, IBM derives substantial revenues from its software patent portfolio. I do argue, however, that patents play completely different roles for small venture-backed firms and for incumbent dominant firms like IBM and Microsoft. The relevant intuition here is that IBM's legendary marketing prowess will allow it to win most contests between reasonably equivalent products. The startup, however, can win those competitions only by

IBM and Microsoft from taking its technology can be the difference between life and death. As one executive put it: "What's protected me from other people ripping [off our product] has been the specter of patent infringement."[113]

It is instructive to think of the offensive uses of software patents reported in the press. Among the most famous incidents plainly are the successful attempts by small firms (Stax in the mid-1990's and Eolas in 2003) to force alterations in Microsoft products that arguably infringed patents held by those firms, and the similar attempt by InterTrust to assert rights to digital rights management technology important to several Microsoft products.[114] The point is consistent with empirical work suggesting that patents held by small firms are more likely to be litigated than patents held by large firms.[115] It also finds strong support in Bronwyn Hall's recent work suggesting that patent rights in complex product industries are more valuable for younger firms than they are for older incumbent firms.[116]

The second point is that the ability of a patent to appropriate the value of an invention will vary along several dimensions. One of the most common is the nature of the particular innovation. Thus, it is considered futile to rely on a patent in which the innovation lies in a method of writing software code.[117] At the other end of the continuum, patents that protect an ultimate functionality that the software provides or an algorithm necessary to provide that functionality are more likely to be important in excluding potential competitors.[118] Interestingly, that distinction seems to undermine the conventional wisdom that "process" patents tend to be less valuable than "product" patents.[119] In the software industry, a patent on the product tends to

---

depriving IBM of the freedom to market a reasonably equivalent product. Thus, the patent's ability to exclude is considerably more valuable to the startup.

[113] Thomas Interview:2.

[114] For discussion of the InterTrust litigation, see http://www.fortune.com/fortune/print/0,15935,400412,00.html. The recent victory by MercExchange over eBay surely deserves honorable mention.

[115] Allison, Lemley, Moore & Trunkey 2003.

[116] Hall 2004.

[117] *See* Denniston Interview:1 ("Is there value in patenting lines of code? Almost never.").

[118] *See* Crouse Interview:1 (discussing range of patent significance at different sectors of Microsoft's operations); D'Eath Interview:5 (contending that his firm's patents create a "competitive barrier," albeit not one of sufficient significance that it would interest investors); Eggleston Interview:2-3 (emphasizing the importance of patents to his company's development, but acknowledging that they are less important for most software companies); Jackson Interview 4 ("If you can get that kind of a business methods patent or a kind of overall process patent for doing things in a certain way, I would say that that's quite important because of your ability to exclude others. * * * * Not having a patent doesn't mean you can't build a business, but having a patent, in my view, certainly strengthens your position."). The perspective of a biotech executive was starkly different (as the data would suggest): "Intellectual property in our industry is the number one reason people fund you or don't fund you." O'Connor Interview:1.

[119] *See* Cohen et al. 2000:10 (empirical finding based on cross-industry surveys that process patents are significantly less valuable than product patents). For a different categorization of software innovation, see Gruner 2000:984-87 (three categories of conceptualization, coding, and external links).

have relatively little value because of the ease of designing a distinct product. A patent on the process that the product implements is much more likely to be valuable, if only because it often is possible for the claimed process to be defined broadly enough to include all practical methods of competition.[120] Thus, to the extent that executives of firms with commercially valuable patents had a view as to why their patents were valuable, they believed that it had nothing to do with the nature of the innovation. Rather, they attributed it to the firm's ability to obtain a patent that staked out a sufficiently large field to cover all plausible variations on the relevant technology.[121]

Related to that continuum is a sectoral variation. As the empirical data presented in the companion paper with Tom Sager demonstrates, there is a strong variation in the rate of patenting among the different sectors of the software industry. For example, the average number of patents in the dataset was about 0.7 patents per firm. Several sectors, however, had markedly higher rates, including graphics and digital imaging, expert systems and natural language, multimedia, and security. At the same time, some relatively important sectors had unusually low rates of patenting, including email and internet software, applications software, and financial software.[122] That variation is interesting, because it can be discerned in a quantitative way even though patents are thought to be less valuable for software than they are for hardware, and even though patents are much less common in the software industry than in some other industries, biotech being the most commonly noted example and the one I analyze. It seems likely that the variation is related at least in part to the nature of innovation in the different sectors, with higher rates of patenting associated with types of innovation more susceptible of appropriation by patent. However strongly my interview subjects rejected such a distinction,[123] the data powerfully suggest that further inquiry is warranted.

Taken together, those two points portray a world in which small firms struggle to innovate, facing the pervasive concern that a competitor might appropriate any useful invention at any time. Given the difficulties those firms face in sustaining differentiation, the likelihood that patents can provide shelter for some firms is an important one. The extent of the shelter may be difficult to predict – because it depends primarily on the breadth of market protection a patent has by the time it is issued.[124] Furthermore, the frequency of the shelter is open to doubt, as it plainly was not relevant to most of the firms that I interviewed. The interviews that I discuss above, however, suggest that it is real in the place where its effect would be most important, in the minds of the firms doing the innovation. It certainly would be valuable to know more about the frequency with which small firms obtain patents of sufficient strength to use in an offensive way. However, even without that information, it is difficult to believe that this is not a major part of what makes patents and their breadth an item of interest to investors.

---

[120] Thomas Interview:2; Urdahl Interview: Hill Interview: Witek Interview.

[121] Urdahl Interview; Thomas Interview; Oles Interview.

[122] The variance in the number of patents each firm has, and in the likelihood that a firm will have any patents, is significant at the 1% level.

[123] *See supra* note 121.

[124] *See* Dinkin Interview.

(b) Indirect Effects

The most intriguing information from my interviews relates to the benefits that patents can provide firms in the industry even if they do not involve direct exploitation of the patents. Those benefits fall into two classes: facilitation of a litigation-free zone through a pattern of cross licensing; and a set of what I call "information" effects – beneficial effects that patents have on information related to the firm's technology.

(I)Facilitating a Licensing Equilibrium

As suggested above, many in the industry completely abjure any substantial use of patents to exclude competitors. Rather, the most prominent explanation in the interviews is that patents will be useful as "barter" in cross-licensing agreements that the firm would enter if it reaches a sufficiently mature stage to be a significant player in the industry.[125] Those with whom I discussed the subject articulated a common vision of the way that firms gain access to protected technology in the industry: they obtain a cross-license from existing industry members that have patents that relate to their technology. To the extent that a growing firm has patents on its own technology, it can reduce the cost of licensing technology from existing market players by providing that technology as part of a cross-license agreement.[126] The likelihood that the firm will be asked at some point in time to enter such an agreement seems to be quite high. Interview subjects acknowledged, for example, that their products might infringe a patent in IBM's large portfolio of software patents.[127] Yet, a patent to offset IBM's potential claim is of little value

[125] Gans, Hsu & Stern (2002) provides a model indicating that this kind of cooperative licensing works better with stronger patents, which is consistent with the more pervasive reliance on cross-licensing in recent years.

[126] *See* Abbott Interview:5 (offering IBM as the example of a potential cross-license partner); Crouse Interview:1-2 (discussing value to Microsoft of building a patent portfolio for defensive purposes); Gauer Interview:2 ("The patent comes in as a defensive mechanism down the road in the event that we stumble upon something else that we've infringed on so that we have our arsenal in order and can use it in a cross-license arrangement or the like."); Kalinoski Interview:5 (describing use of patent to offset request from IBM to pay royalties); Rightmer Interview:1; Subhedar Interview:2 ("You can usually trade, you can usually cross-license if you have patents, so as the company grows bigger, there is value to patent portfolios."); Treybig Interview:1("Patents may give them protection against the bigger company's patent portfolio. I mean, IBM has a room full of patents and it's huge, to the wall, ceiling, and on and on, so, it's somewhat protection against the bigger companies if they have to come after them."); Witte Interview:1 ("And a classic case is IBM will show up and say, "I have this huge portfolio and you must infringe some of mine because I have so many." And then, of course, what you hope is that you have been careful and selected some strategic technology to patent that they may infringe[, so that you can] use that as a shield.").

[127] Rightmer Interview:1 ("IBM probably could sue us on 20 patents if they looked hard at what we do.").

until IBM demands royalties, and IBM usually does not ask for royalties until a firm is earning sufficient revenues to justify the inquiry.[128]

As a matter of policy, it is difficult to know how to evaluate that arrangement. One perspective is that it reflects a classic instance in which sophisticated parties with repeat dealings can reach a state of equilibrium, operating in the shadow of the law with relatively little active conflict.[129] The premise is that the firms with large portfolios will refrain from litigation in a situation of "mutually assured destruction" from competing large portfolios. The destructive capacity of those portfolios is enhanced by the nature of technology in the industry, which could involve dozens of patentable innovations in any single product that a large firm might bring to market. From that perspective, it is natural to believe that litigation occurs most often when the equilibrium fails, when someone other than an active developer holds the patents,[130] or when the developer fails and loses all incentive to cooperate.

More broadly, however, it is difficult to see how that equilibrium can be regarded as a positive benefit attributable to patents. The only benefit that cross-licensing agreements provide is freedom from patent litigation. The cross-license agreements in question provide only freedom of action; they do not involve the disclosure of technology or transfer of any knowledge beyond material on the face of existing patents.[131]

I can postulate several ways in which cross licensing *might* provide net benefits. For example, it might be easier to share technology cooperatively with a patent system rather than without. Most obviously, patents provide a way to discipline firms that are not cooperative; without patents, it would be difficult to discipline firms that seek to use their technological developments offensively. Patents force firms that wish to use their technology offensively to accept the likelihood of suit by other industry players for infringement of the patents held by

---

[128] *See* Abbott Interview:6 (suggesting that IBM only pursues companies "with a certain revenue baseline" and the need to get a patent "before you get on that radar"); Gauer Interview:3 ("A lot of times the company is IBM or somebody big like that. At what point are they going to come after us? Most of the time the answer is that we're too small to bother with, but we have to have a strategy for how to deal with them when it comes up – what might we have to trade them for license rights when we get bigger."); Treybig Interview:2-3 (discussing value of a patent for cross-licensing once you get on IBM's "scope").

For a good example from the hardware industry, consider the early history of Dell, when IBM called seeking royalties shortly after the distribution of an early Dell product. Although those royalties were a "significant hit" to the bottom line, Dell quickly obtained a few patents of its own, which it used to alter the terms of its arrangement to one in which neither side pays royalties. Inman Interview:1.

[129] The seminal discussion is Mnookin & Kornhauser 1979.

[130] Thus, much of the offensive patent litigation in the industry (discussed *supra* text accompanying note 114) is brought by patent holding companies – which have no operating products and exist solely to collect licensing revenues for patents that they have purchased from inventors. *See* RIVETTE & KLINE 2000:135 et seq.; Dinkin Interview:11; Gauer Interview:3; Sikora Interview:1; Subhedar Interview:2; Witte Interview:1; *see also* Burk & Lemley 2003 (characterizing such companies as "trolls"). For further discussion of trolls, see *infra* subpart VI(B).

[131] Hill Interview; Dinkin Interview.

those players. In an industry where innovation is cumulative, where all products include innovations of others, and where it is difficult to produce a product except in a company with substantial funds, cross licenses could be important.

Similarly, patents might provide an effective way to evaluate the value of the technology that each firm has to determine the amount and direction of payment that is appropriate for each cross license. This might be particularly important in an industry, like the software industry, in which there are a large number of players with widely varying patent portfolios. By contrast, in an industry with a small number of relatively equal participants, a straight patent pool (without pair-by pair determinations of value) would make more sense.

Those arguments, however, strike me as post hoc justifications for a practice that at best simply reduces the transaction costs that firms face in gaining access to patent-protected technology. We know little about the terms of license agreements in the industry, or even the frequency and extent to which license agreements involve the payment of license fees. We do know, however, that it cannot be costless to acquire the patents that firms use to enhance their licensing position. If those costs are incurred solely to minimize the costs of the patents that other firms have, and if patents provide no other benefit, then it seems plain that patents would not be providing any net benefit to the industry. Thus, however pervasive it was in my interviews, and however important it is to understanding actual patenting practices, I do not give great weight to the benefits ofcross licensing as a policy justification for patents in the industry.

(II) Information Effects

The last set of effects relate to information generated through the firm's participation in the patent system. Moving along the course of the firm's development, these effects fall into three classes: (A) the ability of patents to facilitate a firm's efforts to codifying tacit knowledge, (B) the firm's subsequent ability to signal the discipline and technical expertise that allowed it to codify that knowledge, and (C) the use of the patent as a signal of the underlying technology.[132]

- Facilitating the Codification of Tacit Knowledge

One of the most intriguing benefits of patents relates to Ashish Arora's recent writings on innovation. He argues that a key problem in transferring knowledge between firms is the ability to convert tacit knowledge – which is difficult to verify or transfer – to codified knowledge, which readily can be evaluated and transferred.[133] Because a patent by definition – at least if it satisfies the statutory criteria – includes the knowledge necessary to enable a person having ordinary skill in the relevant art to replicate the invention, the existence of a patent is strong evidence both that there in fact is substantial knowledge *of some kind* and that the knowledge is not so bound up with the abilities of particular individuals as to be immovable.

[132] For a thorough discussion of the value of patents as signals, see Long 2002.

[133] ARORA, FOSFURI & GAMBARDELLA 2001:95-99.

To the extent that a patent facilitates that process, it can provide real benefits to the firm: codification of knowledge enhances its transferability and thus its value.[134] This analysis has found its way recently into the law review literature as well, in Paul Heald's recent work on transaction costs and patents.[135] More generally, the idea resonates with the notion that productive assets can have no value until they have been brought into a documentary system in which they readily and reliably can be transferred from person to person.[136] Thus, although my interview subjects do not discuss "codification" of knowledge, they do emphasize the importance that patents play in the acquisition of a startup firm.[137] It seems plain to me that there is more work to be done in understanding how patents serve to facilitate the transfer of knowledge – and whether alternate systems could serve the same purpose without the costs of exclusivity. I discuss those questions briefly below,[138] but it is clear that more research would be necessary to formulate any definitive views on the topic.

- Signaling Discipline and Expertise

Once knowledge has been codified in a patent, the existence of the patent itself can send a signal of the skills necessary to obtain it, primarily engineering discipline and market understanding. The premise is that firms that get patents tend to be more careful in their engineering work and to understand what is special about their products better than competitors that do not have patents. For example, one serial startup developer explained:

> [I]n my experience, all a software patent buys you is the fact that you are disciplined in your engineering approach and that it is reflected in your ability to execute technically. Not that it is a means of protection for the investors to believe that you're gonna be the only person that's gonna be able to solve this particular problem.[139]

Those that articulate this line of reasoning generally view the signal as a true one – a plausible indicator of valuable information about the firm that otherwise would be difficult to discern.[140] Notice, of course, that this use of patents says nothing about the uniqueness of the

---

[134] ARORA, FOSFURI & GAMBARDELLA 2001.

[135] Heald 2004.

[136] De Soto 2000.

[137] *See* Beauchamp Interview:5 ("It is a tangible asset that during an acquisition, the investors can hold up to make the argument that they can increase the valuation of the company."); Lee Interview:2. That perspective appears to be justified – to some limited extent – by the attitudes of people at potential acquirers. *See* Crouse Interview:2 (discussing importance of patents in Microsoft acquisitions); Van Arsdale Interview:1 (same); *see* Witte Interview:4 (suggesting that a typical patent would be "asset number 31" in the list of important assets being acquired).

[138] *See infra* subpart VI(A).

[139] Beauchamp Interview:4.

[140] Gauer Interview:2 ("The fact that they were working on something that might be patentable does tend to be tied with them working on new problems and suggests that they are up on the leading edge of people tackling the kind of problems in which we'd like to invest. The patent itself, however,

technology or the ability of the firm to exclude competitors. Rather, it reflects something positive about the ability of the management team to focus and execute. That does not mean, however, that the signal is not taken seriously. As discussed above, many investors think that inadequate market analysis and execution are among the most common reasons for the failure of startup companies.[141]

- Signaling Technology

When the firm reaches the stage at which it is considering acquisition or a public offering, its patents may send a more direct signal of the underlying technology. As the discussion below explains in more detail, larger firms are likely to value patents for reasons quite different from those that motivate small firms: because they facilitate freedom of action by helping the larger company avoid claims of infringement.[142] Thus, investors consider the existence of a patent to play a key role in influencing the "build-or-buy" decision of a larger company: the hope is that the potential cost of patent infringement will make it cheaper for the larger company to purchase the portfolio company rather than build the technology in-house.[143] Even there, however, it is clear that the sophisticated acquirer will focus on the business model that the company has adopted and whether that modelmakes sense apart from the IP that might

---

isn't the point. * * * * The fact that people may have patentable work tends to correlate with working in new areas and being world-class in those areas but I would never make an investment decision based on whether there is a patent or not or whether I thought a patent application would be successful."); Treybig Interview:1 ("[P]atents give you some idea of 'are these smart people I'm dealing with, that want to start this company'?").

[141] The focus on the importance of execution is exemplified by the current New York Times bestseller RAM CHARAN, EXECUTION: THE DISCIPLINE OF GETTING THINGS DONE 2002, a fixture on the desks of startup executives that I have interviewed.

[142] Claims of infringement might be more of a concern for a large firm because its revenue base is so much larger that even a small royalty percentage could result in significant damages. The recent Eolas case against Microsoft is a good example. Eolas received a $500 million verdict for technology that affects, at most, a tiny portion of Windows Explorer. One report calculated that the dispute involved only 305 of the 56 million lines of code in Windows, but the plaintiff still received a royalty of $1.47 for each copy of Windows. Viewed on a pro rata basis, that would suggest a value per copy of Windows of about $500,000.
http://www.boston.com/business/globe/articles/2003/09/22/few_celebrate_this_defeat_for_microsoft/. It is worth noting that the technology in question was invented at the University of California. The patent was licensed to Eolas, but the University of California will receive 25% of the proceeds from the litigation. http://news.com.com/2100-1012-5062409.html.

[143] *See* Beauchamp Interview:5 ("It is a tangible asset that during an acquisition, the investors can hold up to make the argument that they can increase the valuation of the company."); Lee Interview:2; Oles Interview; Urdahl Interview. That perspective appears to be justified – to some limited extent – by the attitudes of people at potential acquirers. *See* Crouse Interview:2 (discussing importance of patents in Microsoft acquisitions); Van Arsdale Interview:1 (same); *see* Witte Interview:4 (suggesting that a typical patent would be "asset number 31" in the list of important assets being acquired).

protect the underlying technology.[144] For example, consider the following comments about Google (a firm that, incidentally, has two patents):

> Do you think the big asset for Google is patents? No, it's a business model that's working and making money. Do you think patents are something they're not being stupid about. They're filing patents and being careful on the off fear that Microsoft might sue them. Do you think they've built into their S-1 or their business plan that they plan on using their patent portfolio directly? No way.[145]

That same explanation can be seen in a much more negative way if it is thought that the patents will not ultimately bring any value to the balance sheet of the acquiring firm. This view on the role of patents in acquisition characterizes them as valuable for "marketing,"[146] convincing the investors in public markets that the company's technology is valuable.[147] The idea is that sophisticated investors at the early stage can evaluate the "true" value of the technology based on a careful analysis of such factors as the company's product, the customer's needs for that product, and the personnel that the company employs to execute its business plan. Thus, the patent has only secondary significance to those investors. Customers or later-stage investors, by contrast, are said to be less willing to undertake such careful evaluations.[148] Thus, the argument goes, they tend to rely (less thoughtfully) on the mere existence of patents in the company's portfolio.[149] That argument is made particularly with respect to protecting the downside in the event of a company's failure.[150] Interestingly, developers often present a similar argument about venture capitalists, arguing that they obtain patents that have no real value to them, in part because they will look good to venture capitalists.[151] The truth of that view in any particular context is of course difficult to assess.

---

[144] *See* Van Arsdale Interview:2-3.

[145] Van Arsdale Interview:3.

[146] In contrast to the "signal" that the patent provides to the sophisticated investor, the value of the patent from this negative perspective is characterized as "optical": something that enhances the appearance of the company but does nothing for the underlying economics or prospects of the firm. *See* Abbott Interview:5. Bartow 2000:8-9 makes this point vigorously.

[147] *See* Weghorst Interview:5 (discussing "marketing leverage" with investors).

[148] *See* Costello Interview:1; Abbott Interview:5.

[149] *See* D'Eath Interview:5 (explaining that his firm's patents would be important to potential "acquisitors" though probably not to venture-capital investors).

[150] *See* Enroth Interview:1 (discussing a "perception that's around that if you have patented something that there's really got to be some level of perceived value there, so if you're liquidating it there is a different level of ability to get some value out of that compared to something that is viewed as not proprietary"); Inman Interview:1 (discussing "residual value" in intellectual property that returned some value to investors when startup failed in the market); Stephenson Interview:1 ("IP is something that in the downside case we can sell off and make something.").

[151] *See* Abbott Interview:5; Rightmer Interview:1.

4. Patents and Large Firms

Although the bulk of my interviewing base is small venture-backed startups of varying sizes, the interviews and publicly available information do provide enough information to give a good idea of the role that patents play in large firms. Because much of the information is plain from the discussion above, I discuss that topic briefly solely to complete the picture.

First, as discussed above, large firms gain *relatively* little from litigation or the exclusion of competitors from patented spaces, because large firms often can compete successfully with small firms without excluding the competitor. The saying that "nobody ever got fired for buying IBM [or Microsoft]" is not baseless, and in a contest between IBM and a small startup, both with equivalent products, IBM (or Microsoft) often will prevail. In contests among large firms, litigation will be rare because of cross licensing of portfolios.[152]

Second, patents will provide considerable benefits to large firms by enabling them to participate in cross-licensing agreements that give them the freedom of action to design and deploy products as they wish, without regard to the IP portfolios of competitors. It may be, as I argue above, that the large firms that use their portfolios solely for that purposes would be better off without the costs of developing and maintaining those portfolios, but in the existing milieu, each firm has a strong incentive to collect patents for that purpose.

Third, many (though certainly not all) large firms will obtain substantial revenues from directly exploiting their patent portfolios. IBM, for example, earns literally billions of dollars each year exploiting its patent portfolio, a significant share of that comes from its software patents.[153] Thus, although different firms have different strategies, for large firms the potential does exist to earn substantial revenues from direct exploitation of patents. Although information about the amount of those revenues would be valuable in assessing the net effect of patents, information on that subject usually is considered proprietary and is not readily available.

C. Summary

Much remains unclear about the ability of patents to induce commercialization in the software industry. For example, although the ability of small firms to use patents to protect themselves is important, it is difficult to tell from the data available how widespread that benefit is. If it is widespread, then it may contribute to the fragmented and highly competitive structure of most sectors of the software industry by providing startups a sufficient time without

[152] That equilibrium may not be stable in the presence of trolls. Recent reports suggest that some (but not all) of the large firms in the industry are engaged in a program to purchase patents that otherwise might find their way into the hands of trolls. It appears that the large firms that declined to participate in that program have reason to expect that they will be asked to pay royalties for using the technology covered by the patents acquired in that process. Conversation with Pamela Samuelson (June 14, 2004).

[153] *See infra* note 206 (discussing software patent licensing by IBM).

competition to commercialize their products.[154] Similarly, it is difficult to disentangle the local effects that motivate firms to obtain patents – as cross-licensing collateral, for example – from the direct effects and the information effects that might provide a justification for the system as a whole. Further, any understanding of those effects must account for the differentiation of their weight as firms progress through the development cycle. Finally, even if those effects elicit funds for the firms that have patents,[155] we cannot be sure that they increase total investment. It is also possible that they simply alter the direction of investment towards patent-protected investments without altering the total amount of investment.[156] Another possibility is that larger firms against whom the small firms' patents are enforced would have invented the same products almost as quickly.

To clarify the overall import of the discussion, Table One summarizes those effects. As that table shows, my re search indicates a basic tradeoff between several effects that are not readily quantifiable. The major burden I discern is the acquisition and use of patents for cross-licensing purposes, which seems to be a deadweight loss for the industry. On the other side are three benefits. The first two largely accrue to smaller firms: the benefits patents provide in sheltering smaller firms, and the information benefits to smaller firms. Both of those are difficult to quantify: the first because it depends on an understanding of whether and to what extent small firms invent that technology sooner than large firms would have invented it without a patent system, and the second because it is almost inherently subjective. The third benefit is the potential licensing revenues that accrue most commonly to IBM and other large firms. That might not be difficult to quantify but, as discussed above, it is not something about which quantitative data is easy to locate.

TABLE ONE
POSITIVE EFFECTS OF PATENTS

	PRE-REVENUE STARTUPS	LATER-STAGE STARTUPS	LARGE FIRMS
LITIGATION RESOURCES	IMPRACTICAL SCARCE	USEFUL AVAILABLE	UNNECESSARY BOUNTIFUL
EXCLUSION BENEFITS	NONE	LARGE	MINIMAL
LICENSING BENEFITS	RARE	OCCASIONAL	VARIED
CROSS-LICENSING BENEFITS	NONE	POTENTIAL	LARGE

[154] This effect resonates with the analysis in Arora & Merges 2004, suggesting that strong IP rights facilitate organization of an industry with independent suppliers rather than integration into an existing firm.

[155] Bessen & Hunt 2003:19-20 finds no relation between patenting and the status of a firm as a new entrant. Because their database involves large firms, however, and because my hypotheses suggest that both new and old firms should have patents – albeit for different reasons – I do not think their findings affect my analysis here.

[156] Moser 2003 provides historical empirical evidence that supports that possibility.

INFORMATION BENEFITS	NONE	SIGNIFICANT	TRIVIAL

IV. Potential Costs of Patents in the Software Industry

A. Patent Thickets

The literature criticizing software patents generally does not discuss the issues related to commercialization. Thus, it does not consider the possibility that the costs of stockpiling patents for cross-licensing purposes exceed the net benefits that the system provides. Rather, it focuses on a single potential problem: the costs that patents impose when they exclude third parties from development. Most famously, Larry Lessig argues that the proliferation of software patents has created an "anticommons" (a term Lessig draws from Michael Heller's work with Becky Eisenberg[157]) or a patent "thicket" (Carl Shapiro's term[158]). This concern also is stated in Jim Bessen's work (by himself and with other co-authors).[159] Specifically, the concern is that there are so many overlapping patents in the industry that potential innovators cannot readily obtain the approvals necessary to conduct their research.[160] The thesis gains some credibility from the nature of software innovation, because, as explained above, a software product might involve dozens of innovations, and several firms might hold patents on one or more aspects of a firm's technology.

I address first the claim of Bessen and his co-authors that the proliferation of patents has stifled R&D spending. I then address the anecdotal claim that a thicket of patent claims deters small firms from pursuing promising innovations.

1. R&D Spending

The two papers are Bessen's unpublished papers with Maskin on *Sequential Innovation, Patents, and Imitation* and with Hunt on *An Empirical Look at Software Patents*. Bessen &

---

[157] Heller & Eisenberg 1998. Formal economic modeling of this problem appears in Buchanan & Yoon 2000; *see also* Parisi, DePoorter & Schulz 2000:17-18 (application of that model to intellectual property).

[158] I use the term from the extended discussion in Shapiro 2001. That discussion is considerably more guarded. He simply notes the possibility of a thicket.

[159] *See* Bessen 2003; Bessen & Hunt 2003; Bessen & Maskin 2002. Bessen's relevant work is unpublished. I respond to it here because it has been widely cited even in its unpublished form. {A search on Westlaw in July of 2004 found 22 citations to Bessen & Maskin 2002.} Kingston and Kash ultimately rest their criticisms on a similar concern, that large firms will "intimidate" outsiders and prevent them from competing. Kingston & Kash 2001:16-17.

[160] Heller & Eisenberg 1998 previously argued that a similar condition afflicts the biotech industry. A contrary view is articulated forcefully by Walsh, Arora & Cohen 2002. For rebuttal, see Rai & Eisenberg 2003.

Maskin compare a dataset of software patents (defined by patent classes) held by large publicly traded firms to R&D expenditures of those firms. They find a correlation between increases in software patenting and declining R&D expenditures. The evidence, however, terminates in 1995 – a very early stage of the software industry – and includes only leading software patentees, rather than leading firms in the industry. Thus, their dataset includes companies like Ford, General Electric, and Japanese firms like Mitsubishi and Matsushita, but excludes companies like Microsoft and Oracle (which were not large software patentees at that time).[161] Furthermore, given the limited importance of software development to the firms that they examine, it would seem important to determine the amount of R&D spending those firms allocate to software development, which they are unable to do with Compustat data.

The more recent paper with Robert Hunt collects a dataset of patents based on key words that are reasonably likely to reflect software innovation. This paper plainly reflects the most serious effort to date to collect a dataset of software patents, as they have collected all patents including the relevant key words from 1976 to 1999, about 131,000 patents.[162] Most of the patents in their dataset are assigned to large manufacturers. Software publishers own only five percent of the patents in their dataset. Their dataset also excludes most private firms.[163] Thus, they do not analyze the firms that generate about one-third of the patents they collected from the PTO database.[164] The primary finding of this paper of relevance here is that the propensity to patent has increased in the software industry since the late 1980's.[165] For the reasons discussed in Part II, that finding is not surprising.[166]

Bessen & Hunt also analyze the relation between R&D spending and patenting, concluding that patents are a substitute for R&D spending. Thus, they argue, R&D spending would be higher if changes in the law had not made software patents more attractive.[167] They conclude that the results are attributable to strategic patenting in a small number of industries that drive their results.[168] Although the data is intriguing, and the analysis considerably more robust than in the Bessen & Maskin paper, it suffers from a number of problems. For one thing, it has several of the same design flaws as the Bessen & Maskin paper, because the focus of the

---

[161] Bessen & Maskin 2002:16-20, 25.

[162] Bessen & Hunt 2003:8-10. As they explain, the word-search methodology produces results that are similar to, though slightly broader than, the methodology John Allison and I are using in our work to identify software patents by direct examination of the patents.

[163] Their sample starts with firms that were listed on CompuStat as of 1989, and is supplemented by adding the 25 largest publicly traded software firms and an unspecified number of private firms based on data provided by CHI. Bessen & Hunt 2003:12-13.

[164] Bessen & Hunt 2003:13.

[165] Bessen & Hunt 2003:21-22.

[166] For a similar result using a dataset that focuses on the software industry, see Graham & Mowery 2002.

[167] Bessen & Hunt 2003:27-34.

[168] Bessen & Hunt 2003:38-40.

paper is not on the software industry, but rather on software patents. As Bessen & Hunt emphasize, "software patenting [as they define it] by and large has little to do with the pre-packaged software industry."[169] Rather, their emphasis is on the other industries in which many software patents are held.[170] Second, as in the earlier paper, the analysis relates firm-wide R&D expenditures – whether or not related to software – to software patents. It is of course possible that software patents have such strong effects that their potential availability might alter the overall R&D expenditures of companies like Ford, but it does seem unlikely. Even for a firm like IBM – which devotes a huge amount of resources to software R&D – the number of considerations that might influence *total* R&D spending that are completely divorced from the software market substantially undermines the credibility of the findings.[171] More broadly, Robert Hahn & Scott Wallsten have identified a number of technical problems with the statistical analysis, which suggest that their core finding that R&D funding and software patents are substitutes is not reliable.[172]

Against those studies – suggesting a puzzling relation between software patents and overall R&D outside the software industry – we can examine data that directly describe the current state of R&D in the software industry. That data suggests a different picture, one in which software R&D is impressively robust. For example, *Technology Review's* Corporate R&D Scorecards report the annual research and development spending of the world's top 150 technology companies. Each company is assigned to one of 12 sectors based on its primary business. The scorecard figures are derived from annual reports and U.S. Securities and Exchange Commission filings. Data from the Scorecards indicates that R&D spending for the software industry is higher than in similar high-tech industries. For example, R&D spending as a percentage of revenues in the software industry for 2002 was 14.5.[173] By way of comparison, R&D spending in the same period was 6.7% for computer hardware, 7.4% for electrical/electronics, and 8.1% for telecommunications. Thus, for the top technology companies, the R&D intensity ratios are high in the software industry in comparison to other industries.[174] National Science Foundation data regarding industrial R&D intensity provides a

---

[169] Bessen & Hunt 2003:38.

[170] Even in the industries for which they identify an effect, the effects are driven by the behavior of a small number of firms. For example, they find a significant effect in the two-digit SIC code 73 (which would include much of the software industry), but acknowledge that the result would disappear if IBM's data were removed. Bessen & Hunt 34 n.40.

[171] Among other things, see Etro 2004 (arguing that parties with monopoly power in industries with sequential innovation may have a *greater* incentive to invest heavily in R&D than outsiders in the industry).

[172] Hahn & Wallsten 2003.

[173] It was 14% in 2001 and 14.5% in 2000. This figure seems to have been quite stable over time. Prominent industry estimates in the early 1980's suggested that the costs of "program development" were at that time about 15% of revenues. CAMPBELL-KELLY 2003:211.

[174] Examples from other countries do not seem to be useful. For example, the software industry in Europe – where patent protection is considerably more ambiguous than it is here – is strikingly underdeveloped compared to the United States. However, it is plain that the relative levels of

similar picture. That data shows that R&D intensity for firms in the software industry (NAICS code 5112) was 19.3%, 20.0%, 16.8% and 20.5%, for the years 1997-2000, respectively, far above the average in all industrial R&D firms of about 3.6%.[175] Indeed, according to the National Science Foundation, the software industry for the past four years has had an R&D intensity substantially higher than any industry other than Scientific R&D Services (NAICS code 5417). Because software development does not depend heavily on the existence of manufacturing facilities and other fixed assets, those high figures should not be surprising. However, it is hard to credit the argument that R&D spending in the industry is systemically depressed.

As this data shows, software R&D spending tends to be relatively stable over time as a percentage of sales. Indeed, the most significant variable in R&D spending within the industry appears to be company size. For example data from CompuStat indicates that median R&D spending for large public companies (over $100 million in sales) in SIC 7372 (prepackaged software) is only 15.9% of net sales, while mid-sized firms (between $30 million and $100 million) spent 22.6%, and small firms (under $30 million) spent 32.8%. Those figures have not changed substantially over the last three years.[176] The Software 500 provides similar statistics for the 500 largest firms in the industry (including both public and private firms). For 2001, firms with more than $100 million in sales had an average R&D intensity ratio of 12.83%[177]; firms with sales between $30 and $100 million had an R&D intensity of 20.49%; and firms with less than $30 million in sales had an R&D intensity of 23.89%. If, patents facilitate a fragmented industry structure by sheltering small firms, they may help to support the high level of R&D spending characteristic of those firms.[178]

The questions that Bessen, Maskin and Hunt raise would be answered best by looking at patenting practices and R&D spending in the software industry. That work, however, is beyond the scope of this paper. For now, perhaps the most that can be said with clarity is the basic point with which I began this section: the patent system is not systematically preventing the initiation of product development. Beyond that, it is plain that the system is functional. In the world that

---

development can be attributed to historical factors that have little to do with patent protection. *See* CAMPBELL-KELLY 2003; Mowery (in MOWERY & NELSON 1999).

[175] NATIONAL SCIENCE FOUNDATION/DIVISION OF SCIENCE RESOURCE STATISTICS, SURVEY OF INDUSTRIAL RESEARCH AND DEVELOPMENT: 2000.

[176] 2001: 16.5%, 26.9, 41.7; 2000: 15.9%, 28.0, 52.4; 1999: 15.1%, 20.0, 41.4. R&D intensity is much higher in SIC 7372 than in the rest of the 7370s (at least for larger firms). The average R&D intensity of the other firms in the 7370 series (in 2002) was 7.10% for large firms, 13.5% for medium firms and 34% for small firms. The average R&D intensity in the 7370 series as a whole was 9.2% for large firms, 17.7% for medium firms, and 38.5% for small firms.

[177] IBM's R&D intensity is far below the industry average (6% in 2001, 5.8% in 2000, and 6% in 1999), although it amounts to over 5 billion dollars each year, while Microsoft's is considerably above the average: 17% in 2001, 16.3% in 2000, 15% in 1999 (about $3.8 billion each year).

[178] I do not mean to suggest here that higher R&D spending is optimal. It is of course possible that R&D spending in the software industry is higher than the optimal level. I only mean to rebut the idea that the proliferation of software patents has stifled R&D spending in any measurable way.

we have – with patents – there are literally hundreds of small firms using institutional financing to develop new technologies. The smaller firms are spending relatively more on R&D than the bigger firms. It is as difficult to be sure that all of those firms would exist if there were no patent protection as it is to be sure that there would not be even more firms if there were no patent protection.

2. Stifling Small Firms

Turning from the empirical evidence about R&D spending to the more fundamental question of industry practice, my interviews and the publicly available information I have located about the industry make it difficult to credit the idea of a "thicket" or "anticommons" in the software industry.[179] When raised in my interviews, that thesis universally was rejected.[180] The premise of the model is that assets will go unused because of the costs of obtaining the permissions necessary to use them.[181] There is of course nothing theoretically impossible about that outcome. The important question, however, is whether this is in fact what has happened in the software industry.

In this case, a few objective indicators suggest reasons why the thesis of the patent thicket has so little ability to say anything descriptive about the industry. For one thing, *none* of the startup firms to which I spoke suggested a practice of doing prior-art searches before beginning development of their products.[182] As far as I can tell, the only occasion in the software industry in which a startup is likely to experience such costs is when the startup is founded on a specific piece of existing technology spun off from an existing company or from a university laboratory.

For another thing, no investor suggested any concern about the possibility that their portfolio firms might be infringing the IP of others in the industry. That is not because they were sure that the startups were not infringing; it was because they thought it would be unlikely to

---

[179] My reactions are based on the history and practices of the software industry. More generally, Rob Merges has long championed the idea that contracting practices often will ensure the effective dissemination of IP throughout an industry. *E.g.*, Merges 1996; Merges 2001:140-46. His current project extends that line of reasoning to private investments in the public domain. He contends that those investments have the potential to limit potential costs from "overpropertization," particularly in the software industry. Merges 2004.

[180] *See, e.g.*, Abbott Interview:6; Beauchamp Interview:6; Eggleston Interview:6; Subhedar Interview:3; Treybig Interview:8; Weghorst Interview:6.

[181] *See, e.g.*, Buchanan & Yoon 2000:4. At its core, the thicket analysis is an analogy to a post-Gorbachev apartment in Moscow that would sit vacant because of the inability of any particular user to obtain consents from all of the various parties with interests in the apartment.

[182] *See* Harlan Interview; Hill Interview: Thomas Interview:1; Urdahl Interview; Witek Interview. Mark Lemley points out that startup firms have a strong incentive not to do such searches, both because the results must be included in later patent applications and because they can affect the determination of willfulness in later litigation. Lemley & Tangri 2003; FTC REPORT:49-50. Conversations with industry professionals suggest that this explanation is consistent with industry practice and with the advice that leading law firms provide their clients.

pose a significant difficulty if they were. As discussed above, industry executives do accept one premise of the patent-thicket thesis: that software patents are multiplying so rapidly that it is likely that many products startups are developing ultimately will infringe patents held by large existing companies. The textbook example is IBM, which apparently holds far more software patents than any other company in the industry. Indeed, as I explained above, several of my interview subjects joked that they thought it likely – without any investigation or particular knowledge – thattheir product would infringe *something* in IBM's portfolio.

Yet that posed no significant concern for those firms. It is perhaps an artifact of the particular history of the industry, but IBM has firmly set a course of *relatively* lenient enforcement of its IP rights. The lenience of its practices is attributed to an attitude developed during its long subjection to government antitrust scrutiny, an attitude of wishing to refrain from conduct that would be likely to interest federal antitrust regulators in its practices.[183] It is now a circumstance long forgotten by many (as IBM is regarded most prominently as a dominant hardware manufacturer), but there was a time when IBM's dominance in the software market was as complete as any dominance it ever has had in the hardware market. Indeed, the most authoritative history marks the beginning of the commercial software industry as the date when IBM began to sell its proprietary software unbundled from its hardware products.[184] If the antitrust litigation tempered IBM's willingness to press its advantages to their fullest, it has limited the rise of a patent thicket in the industry.

IBM's relative lenience also is attributed to the asymmetric risks IBM faces from patent litigation. As the ongoing SCO litigation demonstrates, a finding that IBM's widely distributed products infringe a valid patent is likely to cost IBM much more than a finding of infringement by a small party with a limited customer base that might be involved in litigation with IBM.[185]

That is not to say that IBM allows people to use its IP freely.[186] It is to say, however, that licenses to use its IP are freely available to all legitimate users.[187] Indeed, it appears that the principal, if not the only, reason that IBM would be unwilling to grant a license to use its patents would be if the party requesting the license refused to grant IBM parallel access to the party's own IP.[188] Thus, IBM has followed a consistent two-pronged strategy: attempting to gain as

---

[183] *See* Crouse Interview:3 (discussing history of IBM's patent licensing practices); Dinkin Interview:5-6.

[184] CAMPBELL-KELLY 2003:109-10.

[185] *See* Dinkin Interview:8.

[186] For discussion of the profitability of the licensing to IBM, see GERSTNER 2002:146-52.

[187] IBM's Web site reports that it "is generally willing to grant nonexclusive licenses under reasonable and nondiscriminatory terms and conditions to those who in turn, respect IBM's intellectual property (IP) rights." *See* IBM Worldwide Patent Licensing Practices, *available at* http://www.ibm.com/ibm/licensing/patents/practices.shtml (last visited Oct. 7, 2003); ARORA, FOSFURI & GAMBARDELLA 2001:235-36 (discussing requirements imposed on IBM by consent decrees in antitrust litigation with the United States); CHANDLER 2001:250 (same).

[188] IBM's Web site reports only that it "includes in the terms and conditions of a license an option for a comparable license-back of the licensee's patents under similar terms and conditions," and that "[i]n

much access as it can to all IP in the industry (giving IBM the freedom to market and sell as freely as possible),[189] and at the same time generating a steady stream of revenues from its now massive software-patent portfolio.[190] As Rosemarie Ziedonis has shown, this strategy of heavy patenting is common for large firms like IBM in an environment characterized by fractionation of technology.[191]

The focus on freedom of action is a rational strategy for a large firm like IBM. Considering its advantages in prestige, resources, marketing, and other forms of infrastructure, it is reasonable for IBM to conclude that it can succeed in the marketplace without using the relatively ineffective tool of IP to appropriate the value of its inventions. Thus, the principal relevance of IP to IBM is to ensure that it is able to commercialize whatever products it desires. If the patent portfolio that it uses to ensure that freedom also happens to generate substantial revenues, that is a useful thing, but not nearly so central to the firm's core strategy.

Nor is this strategy unique to IBM. Microsoft, for example, has an impressively large portfolio, but does not appear to enforce it aggressively. Its recent adoption of an open licensing policy that resembles IBM's policy[192] suggests at least an implicit acknowledgment that IBM has discerned the correct strategy. There of course is the possibility that Microsoft's current strategy is motivated as much by its experience with antitrust litigation as IBM's. Finally, other large firms that I interviewed in related industries suggested that their IP strategies were similar.[193]

The noted[194] paper by James Bessen and Eric Maskin articulates a contrary view, reasoning that sequential innovation in an industry with complementarity of inventions is likely to lead to an anticommons.[195] The paper is flawed by its central logical step: reasoning from the wide dispersion of IP rights to the conclusion that IP is not generally available to firms in the industry. The paper does not account for the literature indicating that the effectiveness of

---

cases where a licensing partner has a significant patent portfolio, IBM will consider a patent cross license." *See* IBM Worldwide Patent Licensing Practices, *available at* http://www.ibm.com/ibm/licensing/patents/practices.shtml (last visited Oct. 7, 2003).

[189] *See* Treybig Interview:5 ("IBM['s strategy] is to keep anybody with a patent from hindering what they want to do. * * * * The role of patents was to protect the company against innovation so the company could not be stopped from doing anything it wanted.").

[190] IBM's Web site reports a standard fee of 1% of the "selling price" for use of a single patent in a product, with a maximum fee of 5%. *See* IBM Worldwide Patent Licensing Practices, *available at* http://www.ibm.com/ibm/licensing/patents/practices.shtml (last visited Oct. 7, 2003).

[191] Ziedonis 2004.

[192] http://www.microsoft.com/presspass/press/2003/dec03/12-03ExpandIPPR.asp (last visited Dec. 15, 2003).

[193] I had two such interviews with representatives of Fortune 500 technology firms. Both requested anonymity with respect to that discussion.

[194] As discussed above, the paper is the principal empirical support that Lessig offers for his discussion of the subject.

[195] Bessen & Maskin 2002.

licensing will depend to a great degree on context.[196] Rather, the paper rests on the explicit assumption that firms will not be able to reach value-increasing licensing agreements to make technology widely available in the industry.[197]

To the extent that the literature supports any theoretical conclusions that are independent of historical context, Merges & Nelson argue that a positive outcome is particularly likely – as in the software industry – in cases in which there is not a single pioneering patent or group of patents that gives one firm control.[198] The numerous sectors into which the software industry is divided have made it difficult for any single patent or group of patents to control a major part of the whole industry.[199]

James Bessen's recent paper articulates a more complex model of the same problem. He recognizes the possibility that optimal incentives for research and development can occur when firms develop a culture of "mutual non-aggression."[200] He argues, however, that "aggressive" cross-licensing is a distinct pattern that is likely to lead to sub-optimal incentives for innovation in industries in which patent standards are too low, particularly in cases in which mature incumbents populate the industry.[201] Whatever the merits of that analysis, there is little reason to think that it is applicable to the software industry. As discussed above, it is clear that the licensing culture in the industry depends to a considerable degree on the practices of the industry's leader. In addition, it is clear that IBM has determined for reasons of its own – influenced to be sure by federal antitrust enforcement[202] – that it should refrain from pressing its patent portfolio aggressively. Thus, although it is always possible to imagine that aggressive practices *could* lead to sub-optimal innovations,[203] the historical events that have made IBM and

---

[196] *See*, *e.g.*, Cohen, Nelson & Walsh 2000:25-29; Lemley 1997:1061-67; Merges & Nelson 1990:894-97, 908; Merges & Nelson 1994:13-16; Scotchmer 1991:33-34.

[197] Bessen & Maskin 2002:5.

[198] *See* Merges & Nelson 1990:908-11; Merges & Nelson 1994:4-5.

[199] Nevertheless, there certainly have been important patents that have allowed a single firm to dominate a particular sector for a while. The Rambus patent is the most common example mentioned in my interviews. Urdahl Interview. Interestingly enough, the interviews suggest that the need to patent is particularly high in sectors dominated by such a patent, apparently out of a desire to obtain collateral for cross licensing. Urdahl Interview. It is possible that much of the sectoral variation that appears in my related empirical study is attributable to such historical artifacts. However, for the present purpose the key point is that dominance of a single sector for a time is far from dominance of the entire industry. Moreover, that kind of patent-based dominance has *never* occurred in favor of an incumbent like IBM or Microsoft. Thus, those patents are much less troublesome than, for example, the Edison light bulb patent that Merges & Nelson discuss. Merges & Nelson 1990; Merges & Nelson 1994.

[200] Bessen 2003.

[201] Bessen 2003:14-15.

[202] *See supra* note 187.

[203] The possibility that a disadvantageous structure *could* have developed doubtless explains much of the fears of software patents expressed in the early days of the industry. *See* LESSIG 2001 (discussing such fears); Fox & Kelley 2002 (in FROM IDEAS TO ASSETS):193-94 (discussing those fears, and how they have dissipated with actual experience in the industry). A number of public statements on

now Microsoft as cautious as they are make it difficult to argue that those patterns have emerged,[204] whatever might happen in the future.[205]

What this means for purposes of this discussion is that innovators know that IBM and Microsoft hold large number of patents and that they are likely to use those patents to seek some share of revenues from any major new product. There is, of course, nothing wrong with that. IBM does spend billions of dollars each year on research and development related to the software technology on which it receives patents. It is not alone in that practice.[206] The fact that it can earn royalties from those patents through freely licensing them to all comers does not suggest a patent thicket. On the contrary, a patent thicket would exist only if industry-licensing practices were such that firms in the industry commonly were unable to agree on terms for licenses and thus retreated from the field of innovation. That is not a realistic portrait of the commercial software industry, as it now exists.

Another more plausible possibility is that the "tax" on innovation created by cross-licensing fees is detrimental to innovation in the industry. From this perspective, it is not that the existing patents are so widely distributed as to make it impossible for firms to obtain access to the technology, it is just that the cost of paying for access to that technology lowers the return on investment in the industry to the point that investments in innovation are less than they would be without patents. Part of the difficulty in assessing that possibility is the intractability of determining whether a typical 5% license fee is a sufficient drag on a small firm's profitability to amount to a substantial burden. The problem would be more severe if firms often had to pay multiple licensing fees, but that seems uncommon based on the interviews that I have conducted.

---

that point were delivered in a 1994 Department of Commerce hearing in San Jose, at which several firms (including Adobe, Borland, and Oracle) argued against software patents, while several others argued in their favor (including IBM, Intel, and Microsoft). The statements are available at http://lpf.ai.mit.edu/Patents/testimony/statements/.

[204] For a typical news story, see *Patents: A Necessary Evil* 2002, *available at* http://news.com.com/2009-1001-801896.html (last visited Oct. 2, 2003) (discussing IBM licenses that permitted development of technologies for DVD, PalmPilot, Handspring, and TIVO).

[205] As I suggest *supra* note 152, there is some possibility that the equilibrium I describe here may change in the future. It remains to be seen, however, whether it will alter the basic pattern, in which the smallest startups are left relatively free to develop their products without constraints from the patent portfolios of large firms.

[206] *See also* Crouse Interview (suggesting that the ability to generate patent royalties from Microsoft's research department helped justify that section's budget allocations). It is difficult to get details about revenues from software patents, but overall patent licensing revenues have risen from less than $20 billion in 1990 to well over $100 billion by 2000. RIVETTE & KLINE:6. At IBM alone, IP licensing revenues were more than $500 million in 2002. IBM does not report separately the figures for software-patent licensing, but it has reported that about a third of the patents it has received in the last decade (7,500 out of 2,500) and the last year (12,50 out of 3,300) are software patents (*IBM Tops U.S. Patent List* (Jan. 13, 2003), *available at* http://www-3.ibm.com/software/swnews/swnews.nsf/n/mmaa5hrqgp?OpenDocument&Site=default (last visited Oct. 7, 2003), so it is reasonable to estimate that its software-related patent licensing revenues are in the range of $150-$200 million a year.

In the end, my instinct is that it is not a substantial burden. I am driven particularly by the point that the licensing fees normally are paid only on revenues – not simply on use of the patented technology – and thus impose no costs on firms that are in a pre-revenue development stage.

B. Patents and Open Source Development

In my mind, the biggest question about the effectiveness of software patents relates to the interrelation between commercial software development (the topic of this paper) and open source development. This paper relates almost entirely to the commercial software industry, where software is developed and commercialized in an institutional way. My evidence suggests that within that framework patents are useful, largely because they offer more benefits than costs to small firms. However, coexisting with the commercial software industry is a large and apparently growing open source community, which develops software largely without commercial investment or affirmative IP protections.[207] Those who work in that community may have little need for patents. The cooperative nature of development obviates any need for the actual and implicit cross licensing that disseminates access to technology throughout the commercial software sector. Similarly, because open source developers do not depend on outside equity investment to any significant degree, the limited ability to appropriate a software invention poses little harm to them.

The problem, however, is that the open source community does not exist in a vacuum. It exists in a world in which the commercial software industry is building up large portfolios of protected IP, portfolios that pose a serious threat to open source development. To put the matter in a current context, suppose for a moment that the Linux operating system in fact infringes in a substantial way patents held by some major proprietary software firm. That could result in liability for all of the many firms using the Linux operating system. The problem is that the open source community has set itself outside of the cooperative IP framework of the mainstream software industry. Thus, its members often have no patents of their own with which they might protect themselves in such litigation. At the same time, this community has developed its software with the same cavalier attitude to the possibility of patent infringement as commercial software firms exemplify. Those two habits cannot coexist in the end.

That raises the question, in turn, whether the potential for high-quality software development through the open source modeljustifies eradication of software patents even for the commercial software industry. To put it another way, one potential cost of permitting ready enforcement of software patents is the potential disabling of the open source model.[208] It is

---

[207] The literature on that subject is large and contentious. The classic source is Raymond 1998. For a lucid, accessible, and reasonably balanced introduction, see FINK 2002.

[208] Zittrain 2004 discusses the difficulties that the open source movement faces in responding to attacks from proprietary firms (like the SCO litigation) alleging that open source code has been contaminated by copying proprietary software code. My point here is a more systemic one that cannot be avoided even by thorough review of code incorporated into an open source project: the open source project would infringe the patent even if the open source code were created independently, without any knowledge of the patented technology.

difficult to answer that question definitively without evidence that would allow a comparative weighing of the benefits of open source development against the benefits that the commercial software industry derives from IP. It does seem relevant, however, that the reluctance of the open source community to obtain patents is largely a political statement, not something necessary to the development of the improvements in functionality that the open source movement promises.[209] Indeed, the prominent use of the copyleft technique is a direct rejection of the notion that software technology should be free from constraints imposed by its creator.[210] This is proven most clearly by the recent applications for patents filed by noted Linux distributor Red Hat.[211] To the extent the open source community is put at risk by the proprietary community solely because of moral distaste for patents, the claim that the proprietary community should not be able to use patents to advantage its own operations is weakened.

In any event, a thorough analysis of that question is beyond the scope of this paper. Among other things, such an analysis would have to account for the rapid convergence of commercial and open source licensing models – so that even proprietary licenses now commonly allow access to source code and purportedly open source licenses regularly permit commercial development of proprietary works derived from the covered products.[212] I note the issue here only to define the bounds of my analysis. The primary goal of this paper is to consider the role of IP in the commercial software industry.

V. The Role of Other Existing Systems

If the ultimate question is whether patents facilitate the ability of software firms to appropriate the gains from innovation, the picture must include some understanding of the alternate methods that firms might use to appropriate the value of inventions. Those alternate methods are particularly important given the clear evidence that in many circumstances patents will do little to allow a firm to exclude competitors from a firm's innovation. This Part discusses the two main existing legal systems that complement the protections afforded by patents: copyright and trade secrets.

A. Copyright

Like patent protection, the role of copyright protection changes markedly as the firm develops. My interviews suggest that copyright is of relatively little value in protecting the startup from competitors. It does have value, however, in two particular circumstances:

---

[209] For a clear discussion of the distinction between the moral and pragmatic segments of the open source community, see Zittrain 2004. Merges 2004 views investments in open source development as an example of property-preempting investments – investments designed to create a field of innovation from which IP exclusivity will be absent.

[210] *See* McGowan 2001; Zittrain 2004.

[211] Broersma 2002.

[212] *See* MANN & WINN 2005 (discussing those developments).

preventing piracy at a company's later stage when it has developed a product; and preventing "theft" of materials by outgoing employees.

1. The Role of Copyright in Startups

For purposes of this paper, the key question is the extent to which copyright protection can provide the kind of sustainable differentiation that is important to investors. The preceding Parts display considerable ambiguity on the ability of patents to provide that differentiation. With respect to copyright, however, the question is much less ambiguous. Rather, it seems quite clear that copyright is not suited to providing that protection. The problem with copyright protection for software is that the legal system for copyright is not designed to protect functionality. Because functionality in most cases is the aspect of software products that makes them attractive to customers,[213] the differentiation that is important to investors is differentiation in functionality. Thus, if the legal system works as designed, copyright should be useless at this point.

My interviews strongly supported that perspective. For example, a typical startup executive explained that copyright protection "is not useful to us [because of its inability to protect functionality]. The other person could do just the same thing in a different manner and get around it very easily."[214] Another argued: "I've been in the software business for 20 years. Copyrights are worthless. They are totally worthless."[215] One thoughtful executive opined:

> Copyright solves one problem, which is the whole or partial copying of an expressive application. The whole or partial copying of an application by a pirate you can get. But it doesn't really protect us from sharing our technical information broadly and a company then understanding how our products work.

---

[213] There are, of course, types of software for which functionality is *not* of central importance. Video games, for example, are software products for which the expressive content is the primary market differentiator. What that means for my purposes is that the relevant IP protections for video games should look much more like those for traditional audiovisual works (such as motion pictures). Knowledgeable industry executives recognize this distinction as crucial in the negotiation of contracts related to the exploitation of those works. *See* Koontz Interview:1.

[214] Harding Interview:1; *see* Beauchamp Interview:8 ("To what extent does [copyright] keep people from stealing your ideas or your product? None."); Costello Interview:1 ("Generally, I think that most people in the software industry don't think it is worth all that much."); Kalinoski Interview:8) ("I've seen enough of copyright litigation and the issues with copyright law that don't have any real bite. They don't have – * * * there is no impact. * * * [P]eople think that by having copyright * * * that no one can really copy things * * * and it just doesn't stand up in the marketplace."); Kelley Interview:1 ("What really it does come down to is what is the thing that is marketable and if this thing that is particularly marketable is functionality, then the patent is clearly playing a more important role."). For similar comments from venture-capital investors, see E. Jones Interview:2 ("We don't make copyright a big issue."); Lee Interview:1 (copyright is "not a focus area for us" because it is "not useful"); Murphree Interview:1.

[215] Adams Interview:1.

> Patents are inter-industry mechanisms for creating value. Copyright is creating protection between the industry and the channel or end customers.[216]

The most obvious problem with that argument relates to reverse engineering.[217] Generally, the expression that copyright protects in computer software is in the lines of code of which the program consists.[218] Thus, although copyright does not prohibit a competitor from writing a completely new program that includes the functionality of the existing program, it does bar a competitor from taking the existing code to produce that program.[219] Thus, the effect of that protection turns on an empirical question: how effective as an exclusionary device is it to require a competitor to rewrite a competing program instead of reusing the code?

Surprisingly, my interviews indicate quite strongly that it is not effective. The perception is that in most instances a software engineer that could observe the program in operation could readily understand the functionality that the software provides and with that understanding easily could write code that would provide the same functionality: "[S]oftware in general is very malleable and is easily reverse engineered."[220] As one venture investor explained: "The

---

[216] Kaefer Interview:2. For a similar emphasis on the vertical – rather than horizontal – value of copyright protection, see D'Eath Interview:6 ("If somebody goes and takes the actual code that is a pretty stupid way of competing. No valid competitor is going to just take the product and steal the code."). As I discuss below (Section V(A)(2)(a)), copyright's role in preventing piracy is arguably its most important role in the software context.

[217] There is a terminological complication in the discussion that follows. To some, "reverse engineering" has a narrow meaning that implies an effort to duplicate existing source code precisely. *See, e.g., infra* note 220 (discussing sources that take that perspective). In most of my interviews, however, it was plain in context that "reverse engineering" referred more broadly to an effort to recreate functionality, without regard to recreating existing source code.

[218] *See, e.g.*, Karjala 1997:72-77. There also of course is a literary work protected by copyright in the interfaces through which users interact with programs. And in some cases at least, as with the video games discussed above, it is plausible to think that the interface itself could be important to the market success of the product. To the extent that is true, copyright protection for the interface (which is relatively thin for the reasons discussed above) would have the potential to exclude competitors by making reverse engineering illegal. As the text suggests, that seems to me an inappropriate application of copyright doctrine. *See* Karjala 1997:75-77, 94-112 (arguing that copyright protection does not extend to software interfaces).

[219] *See, e.g.*, Karjala 1997:72-75.

[220] Weghorst Interview:3. This is consistent with the understanding of Pamela Samuelson and her co-authors (including the noted software engineer Mitchell Kapor), who contend that the know-how of software is for the most part "near the surface" and readily extractible through testing. Samuelson, Davis, Kapor & Reichman 1994:2317-20 (arguing that programs with different code but identical behavior are market substitutes). *But cf.* Abramson 2002:128 (expressing doubt about utility of "black-box testing"); Strasser 2001:23 (similar perspective). Abramson and Strasser address reverse engineering designed to recreate the existing source code. I share their view that it would be difficult to do that. As the text and Samuelson's discussion recognize, however, it is clear that the goal of the typical reverse engineering process is not to reproduce the existing source code as much as to understand and reproduce the functionality that the source code effects. *See supra* note 217. I see no reason to doubt the value of "black-box testing" for that task.

difficulty normally is managing the people, not solving the problem. The code won't look the same, but the functionality will."[221] One developer explained that the difficulty in coming up with a successful enterprise software product is not writing the code, but understanding the problem that needs to be solved.[222] Thus, the only IP protection that would make it difficult to duplicate a program's functionality would be patent protection – which would bar a competitor from writing code that includes any patented aspects of the software's functionality. If copyright systematically stops short of providing that protection, it cannot provide the vehicle for appropriating value that would persuade potential investors.

2. The Role of Copyright in Later-Stage Firms

Yet, it is plain that copyright plays a crucial role in the industry's ability to appropriate returns from the innovation that it produces. That role has several aspects, but two are sufficiently pervasive that they can be characterized as structural: the prevention of piracy and the control of "theft" of code by departing employees and the like.[223]

(a) Piracy

On the first point, the discussion above explains that copyright protection is unimportant for the startup firm because literal copying of the code is not that important to the competitors of the startup. There is one group, however, that would be quite interested in a free right to copy

---

[221] Lee Interview:1; *see* Murphree Interview:1; Stephenson Interview:1 ("Copyright only protects the particular source code, the instantiation, the physical lines of code that you wrote. And in software there are a number of different ways to accomplish the same thing."); *id*. ("And plus, the bigger point is that all you really get is protection on that particular instantiation. If they change * * * and they're not violating that copyright, then it's not really doing that much good. Because ultimately what you want to protect is the functionality."). For similar views from developers see Bishop Interview:2 (arguing that reverse engineering is easy when the ideas in a product are "what the customer wants"; Like [graphic user interfaces], once you know that's what people want, easy to do."); Costello Interview:1 ("[T]he general problem [with copyright protection] is there are a million ways to go around it. And I think that is true for almost all IP in software."); Kalinoski Interview:9 ([Your competitors a]re at least smart enough to go modify and have something that's VERY * * * similar. It's got the same attributes. It's got a lot of the same qualities, but it looks just a little bit [different even if] it's obvious where they got all of their thinking."); Rightmer Interview:2) ("Reverse engineering is easy. Copyright protects only against blatant theft. Ever since Lotus lost protection for look and feel, copyright has not seemed valuable in the industry. We do this all the time for our customers, to mimic the functionality, input, and output of existing products that we are replacing.").

[222] Weghorst Interview:4 ("[T]ypically, when you're talking about enterprise software, * * * the magic dust, if you will, is in the domain expertise [of] knowing that we need to solve the problems that we needed to solve.").

[223] Data from the Northern District of California for 2002 show that 27 out of the 90 copyright cases filed involved software. Of the 22 software-copyright cases for which I could obtain a complaint, the majority (15) involved claims of unauthorized distribution of the copyright owner's product. Four involved claims against competitors for wrongful reverse engineering, and three arose out of contract disputes between web-site developers and their customers.

the startup's product: its customers. Thus, where patent and trade secret protection are more important in limiting the ability of competitors – horizontal firms – to take technology from the innovator, copyright is more important in limiting the ability of customers to obtain the product without paying the product's owner.[224]

That problem – piracy, to use the industry's preferred term – affects different types of software differently. For example, it is less important in the enterprise software market in this country. To be sure, there are reliable methods of limiting piracy in that market. In some cases, firms operate as application service providers, so that the code for their program resides entirely on their own server, which can be protected more readily than the servers of their customers can.[225] Other companies, particularly in the enterprise software sector that is the source of most innovation in the industry, emphasized the practical value of dealing only with large and fiscally responsible "Fortune 500" customers.[226] A typical example: "[W]e're selling to an enterprise customer. We're not on a store shelf. So I'm not at risk of somebody copying the disk and just cloning what I do."[227] Key to the effectiveness of those arrangements is the likelihood that the customers will be large and creditworthy firms. Those types of firms are unlikely to participate in illicit distribution of software code, if only because of the likely financial exposure they would incur if their participation were discovered.

In some markets, however, those protections are not useful. Most obviously, they are not valuable in markets (such as consumer markets) in which software code (in any form) is freely distributed.[228] Even in this country, it is apparent that consumers commonly violate the terms of license agreements, copying and transferring software in ways that would require payment from the new user.

Executives recognize that in other countries the problem is a serious one even in the enterprise software context.[229] For reasons that range from an intentional governmental design to foster piracy to mere lackadaisical toleration of piracy, the extent of piracy in many foreign countries is shocking: industry statistics indicate that in many countries less than 20% of the

---

[224] *See supra* note 216 and accompanying text (articulating a distinction between vertical and horizontal protection).

[225] *See* Harding Interview:2.

[226] *See* Bishop Interview:3; D'Eath Interview4; Rightmer Interview:2.

[227] D'Eath Interview:6.

[228] *See* D'Eath Interview:7 ("That answer is going to be very different for somebody who is selling a retail product."); Kelley Interview:2 (emphasizing importance of copyright protection for mass-market software); *see also* Karjala 1997:67 ("[O]nce * * * programs are distributed in object-code form, they can be copied almost without cost in large numbers.").

[229] *See* Adams Interview:1 (discussing piracy of Lotus in Italy); Cranton Interview:1; D'Eath Interview:7 ("Because [if you sell overseas] you then in fact could have somebody just copying it, in France or Germany or somewhere where they're not watching."); Levins Interview:1) (discussing major businesses engaged in piracy in Eastern Europe); Sikora Interview:1 (offering examples of Russia and China); *see also* Cranton Interview:2 (suggesting that enforcement of copyright protection over the Internet is similar to enforcement in a developing country).

software actually distributed is accomplished through lawful channels.[230] To be sure, the piracy in those countries is not a total loss to the software developer, because it helps to develop a network of users that make the product more attractive to others. Still, it does reflect a loss of revenue that the software developer could obtain if its IP rights were enforced effectively.

For several reasons, copyright is the only effective IP protection against piracy.[231] For example, even if the pirated software were protected in part by a patent, a suit against the pirate challenging patent infringement necessarily would be more difficult, because of the need for the software owner to establish the validity of the patent. Because of the high standard of patentability, it always will be difficult for the patent owner to get over the threshold of patentability.[232] Because of the low threshold of copyrightability, it never will be difficult for the owner of copyrighted software to establish that the software includes copyrightable innovation.[233] In addition, the limitations on copyright protection discussed above – which make copyright useless for the startup trying to protect the functionality of its software – will be irrelevant in the case of the pirate: the pirate by definition will have copied all or substantially all of the product. Those problems are particularly important in the enforcement of criminal sanctions for piracy. Statistics from the Department of Justice suggest that the federal government often sues pirates for criminal copyright infringement; there is not a statute for criminal patent infringement.[234]

---

[230] *See* IDC 2003 (reporting piracy rates of 94% for Vietnam, 92% for China, 88% for Indonesia, and 87% for Russia and for Ukraine). Nor is the problem limited to Asia and the former Soviet Union. The IDC data suggest that piracy also is rampant in countries in Europe (Bulgaria and Romania 75%, Croatia 67%, Greece 64%), the Middle East (Kuwait 76%), and Latin America (Costa Rice 64% and Argentina 62%).

[231] *See* Crouse Interview:1 ("[C]opyright remains incredibly important for us [at Microsoft]. Without that piracy on a worldwide basis as a form of competition – it would be hard for people to sustain a business against."); Karjala:1997:67 ("Because the evil to be avoided was * * * slavish copying, especially slavish electronic copying, because copyright protects at least against *that*, and because computer programs formally fit the broad definition of a literary work under copyright law, it became a natural candidate for the protection of programs, notwithstanding their inherent functionality."), 69 (arguing that "protection against piracy" should be the "policy goal of software protection under copyright"); Samuelson & Scotchmer 2000:1613 ("Copyright law protects programs from the cheapest and most rapid way to make a directly competing identical product, namely, copying program code exactly.").

[232] *See* 35 U.S.C. §§ 101-103 (principal conditions for patentability); TOEDT 2002:§§ 3.3 - 3.4 (discussion of patentability requirements as applied to computer software).

[233] *See* Copyright Act § 102(a) (extending copyright to "original works of authorship fixed in any tangible medium of expression"); SCHECHTER & THOMAS 2003:§ 3.1.2 (discussing "minimal creativity" requirement for copyright protection).

[234] For example, DOJ data shows 25 indictments of 73 defendants for criminal copyright infringement in 2002. http://www.usdoj.gov/ag/annualreports/pr2002/AppendixC.htm. For similar views from a former prosecutor, see Levins Interview:1); *see also* Rubin Interview:2 (similar view from executive at Microsoft).

(b) Pre-Market Protection

Copyright also plays a role before a product goes to market, in helping a firm prevent technology from leaking out through the actions of employees and business partners. The obvious problem is policing the activity of departing employees. It is common in all startup sectors – including the software sector – for employees rapidly to move from firm to firm. Indeed, Ron Gilson argues with considerable force that California rules limiting the ability of firms to prevent those moves are crucial to the success of the venture-capital industry in Silicon Valley.[235] Yet, it is one thing for employers to allow the cross-pollination of employees and their human capital and experiences. It is another for their employees to take substantial pieces of "product" out the door with them and sell that product from their new company.[236] As Rob Merges explains, it surely is optimal for firms to have *some* control over that activity, and it is not plain that the parties can protect themselves adequately through contracts alone.[237]

Thus, although the discussion in the next section evinces considerable skepticism about the social value of robust trade secret protections, my intuition is that copyright protection serves an important function. In this context, patent protection is relatively ineffective because of the litigation costs and uncertainty of such litigation. But in cases in which the employees attempt to reuse a substantial amount of code from their previous firm,[238] copyright law provides a simple and effective remedy against the new firm.[239] The importance of that constraint in the system is evident from the discussions of corporate counsel about their diligence with respect to new employees[240] and from venture capitalists about their investigation of potential investments. The only instance in which I heard venture capitalists express concern about preexisting IP of other firms constraining the ability of their potential portfolio firms is the case in which a startup has engineers with previous experience designing a similar product, which raises the risk of code

---

[235] *See* SAXENIAN 1994; Gilson 1999.

[236] *See infra* note 248 (suggesting a distinction between employees departing with patented or copyrighted technology and employees departing with other forms of trade-secret information).

[237] Merges 1999.

[238] "We have situations all the time during the course of development where our code walks out the door. Or we have rogue employees or contractors who have access to the code who leak it." Rubin Interview:2; *see* Beauchamp Interview:9 (suggesting that copyright's primary value is as "a protection against possibly disgruntled employees or somebody that may have access to the source code").

[239] The injunctive and criminal remedies discussed above also are important in that context. *See* Rubin Interview:2; Witte Interview:5.

[240] *See* Witte Interview::

> [T]he way that really happens is when an employee moves from one person to another person, and takes the code with them. The thing that you are most worried about is hiring somebody as an engineer to build code, and who in the interest of moving from Point A to Point B in the most efficient way just borrows some of what they had from their last employer. * * * * I worry a little bit about somebody taking out, but if you think about it, if somebody leaves my company and goes to join Microsoft and stuff finds its way into Microsoft – I've won the lottery!

contamination.[241] Although state-law causes of action based on misappropriation of a trade secret, unfair competition, or breach of employment agreements might remedy some of those problems, the clarity and simplicity of the copyright action, the ready availability of federal jurisdiction, and the statutory remedies combine to make it a significant tool in policing such conduct.

A similar problem occurs for large companies engaged in collaborative development projects. In that context, the copyright protection that attaches during the development process is an important part of preserving exclusive rights to the code as it passes among the participants in the process.[242] Although the participants in that process are free to use contracts to define the rights each of them has in the various portions of the project, the lawyers that participate in that process argue that the injunctive remedies and statutory damages available under the Copyright Act play an important role in establishing a robust enforcement backdrop for those arrangements.[243] As one executive explained, "[t]he protection of [a major new product], as it is being designed, and built, and tested, and being distributed to third parties, is critical. And anything that diminishes the protection of that pre-release code will impede our ability and willingness to get outsiders to look at the code and test that."[244]

B. Trade Secret

Trade secret protection plainly plays an important role in the software industry, if only because it is clear that many companies have no formal IP protection for their products[245] and take significant steps to keep the details of their technology secret.[246] There are limits, however, to the extent that trade secret protection can provide a robust appropriability mechanism in the software industry.

---

[241] One venture capitalist explained:

> The only time there is a concern [about copyright] is if you have a team that has come out of another environment, like Sun, we're concerned about having a free working environment up front. If we thought it was encumbered in some way up front, it would be a problem. * * * *
>
> It is fairly frequent that we're asking the question: Is there any code at all from your old employer that is in this. And frankly it is always in the way and there is never a clean way of doing due diligence in this other than getting to trust the people not to have walked off with some of this.

Gauer Interview:1.

[242] *See* Rubin Interview:1.

[243] *See* Rubin Interview:1-2 (discussing Copyright Act §§ 502, 504).

[244] Rubin Interview:2.

[245] In my dataset of software companies that first received venture-backed financing in 1998 and 1999, only 20% of the companies (152/778) had received a patent by the end of 2003. More than half (51%) of comparable biotech firms had received a patent by the same point in time.

[246] *See supra* notes 225-227 and accompanying text.

First, as suggested above,[247] my interviews strongly suggest that it is easy for competitors that observe a new product to design and deploy products that include the functionality of that new product. Such conduct does not violate trade secret law. More broadly, trade secrecy does nothing to provide the "foothold" protection that is useful for smaller firms trying to fend off large-firm efforts to market competing products.[248] In an industry in which innovation involves many firms trying to do the similar things at the same time, it is likely that a large firm like IBM or Microsoft might make the same advance that a small firm has made, even if that large firm has no access to the small firm's technology, and thus no responsibility under trade secret law. Although patents arguably give small firms some shelter in those contests, trade secret law does not offer the same protection.

Finally, as suggested above, there is some reason to think that vigorous enforcement of trade secret protections in some contexts – against departing employees, for example – is itself costly to the industry. Saxenian and Gilson have presented a rich descriptive account arguing that the development of a rich innovative culture in Silicon Valley depends in part on the free transmission of noncodified information[249] by employees moving from firm to firm.[250] Given the picture of cross-pollinating innovation I describe in this Article, that effect should be particularly valuable in the software industry.

VI. Alternative Systems

I do not intend to propose a new system for IP protection in the software industry.[251] Indeed, our international treaty obligations would make it difficult for us to substitute any such system for the systems now in place.[252] I do think, however, that it is useful to explore in a summary way the possibility that some reform short of a full-scale abolition of patents might readily solve the problems that patents cause while leaving in place the benefits they provide. My discussion is frankly skeptical in tone, reflecting my view that it is difficult to be sure that any intervention would improve the existing system.

---

[247] *See supra* notes 220-222 and accompanying text.

[248] *See supra* section III(B)(1).

[249] I do not explore the question in detail here, but it seems to me that it is reasonable to draw an upper limit that cuts off the free transmission of that information at a point where it has been codified into a patent or specific software code subject to copyright protection. The harms to the firm that loses that technology seem more serious than when the loss is in the nature of knowledge that the firm has not yet developed into a specific implementation. In other words, the benefits of free transmission are enhanced when nothing has been done with information or knowledge, because the likelihood that the new firm will produce something the first firm has not is enhanced when the first firm has not done anything sufficiently specific to warrant copyright or patent protection.

[250] *See* SAXENIAN 1994; Gilson 1999.

[251] A thoughtful and well-informed group of scholars already has undertaken such a project, resulting in the justly prominent *Software Manifesto,* Samuelson, Davis, Kapor & Reichman 1994.

[252] *See supra* note 13.

A. Registration

The direct benefits of patents necessarily depend on the right of the patentee to exclude competitors. That is not so clear, however, with respect to the indirect benefits that I discuss. In particular, the indirect benefits that seem to produce social value[253] depend entirely on a range of information effects that do not require that the information that has been articulated and codified be deployed to exclude competitors from any particular product they in turn might wish to deploy. This suggests, in turn, that those benefits could be obtained through a simpler registration system, in which the technology is registered with a central office that simply receives the filings but does not evaluate them for novelty or obviousness.[254]

Although such a system theoretically might provide the indirect benefits discussed above, while avoiding some of the potential costs, it has a number of obvious problems. First, it would be difficult to replicate in a less formal process the benefits that come from the information provided by patents in the current process. For example, the benefits from codification of knowledge may not depend on an exclusive right to the knowledge, but they do depend on the insurance that the knowledge has been reduced to a sufficiently precise formulation that it can be patented. It is not clear how a private office or a simple registration office (like the Copyright Office) could provide those benefits.

Second, to a lesser degree those benefits depend on the possibility that the patents at some later date in the development of the technology will have the potential to exclude people from the technology. I think of the frequent suggestions in interviews that patents are valuable in making a firm attractive for acquisitions. Although the smaller firm with those patents may not be using them to exclude the potential acquiring firm from those products, the value to the larger firm may rest in part on the exclusive potential of those patents.[255]

Finally, what we know about the behavior of software firms suggests that the implementation of such a system might face considerable practical difficulties. We know that firms in the current system strongly resist registering their software with the Copyright Office, even in circumstances where such registration would facilitate lending (by enhancing the ability of a lender to obtain an enforceable security interest). As I have discussed, the reticence of young software firms to disclose their technology makes them reluctant even in the face of the significant benefits in facilitating financing.[256] Here, the costs of disclosure presumably would

---

[253] I argue above that the private benefits of cross licensing do not reflect a social benefit: that all in the industry would be better off avoiding the costs of accumulating those stockpiles.

[254] This discussion builds on a conversation with my colleague Oren Bracha. The suggestion calls for something akin to the German utility model system. *See* Reeves Bros. v. United States Laminating Corp., 282 F. Supp. 118 (E.D.N.Y. 1966); Bleistein (1937).

[255] Of course, the value also may rest in part on the freedom to act those patents provide in the patent-stockpile scenario. That value, it seems to me, should not be part of this calculus.

[256] *See* Mann 1999.

be greater than in the Copyright System,[257] because the disclosure actually would reveal the technology.[258] On the other hand, the benefits would be considerably less tangible: the ability to show potential investors that the firm has, and understands that it has, discrete technological advances.

In sum, the information benefits I discuss above are likely to be so intertwined with the existing patent system that it would not be practical to design an alternate system that could disaggregate them from the exclusive rights to technology that characterize the existing system.

B. In Praise of Trolls

The final possibility is some form of compulsory licensing to mitigate the costs of breakdowns in the licensing equilibrium. As discussed above, a significant portion of the scenarios in which suits are brought in the industry as it presently exists involve what Mark Lemley has christened "trolls," small litigation-oriented firms that exist only to exploit patents, not to develop products. Because those firms do not develop products of their own, they need not fear the costs of countersuits by defendants claiming that the products of the troll infringe patents of the defendant. Thus, the risks of patent litigation for trolls are considerably lower than they are for the typical operating software firm. A natural response to that analysis would be to limit the right of trolls to enforce their patents in some way. An obvious possibility would be compulsory licensing – in which a third party would set a "reasonable" rate at which the patent must be licensed.

I am dubious that such a proposal could be implemented in a way that unconditionally would increase incentives to innovate. For one thing, it is difficult to contemplate a proposal that could define a relevant disfavored class of patent plaintiffs and limit their royalty rights to some compulsory license fee. Although the suits of "trolls" frustrate many in the industry, any effort to design a suitable definition of the term "troll" is likely to lend credence to the view that the status as a "troll" is in the eye of the beholder. Every firm that has a patent that is sufficiently valuable to support major litigation against a large firm marketing a product that arguably infringes the patent has acquired that patent from some person that has invested the resources to invent that technology. It is difficult to discern any principled distinction between the desire of that inventor to appropriate the value of his invention and the desire of operating firms to appropriate the value of their inventions.

The fact that the invention may have been assigned by the inventor to a third party does not suggest that the right to enforce the patent should be diminished.[259] To use an example from my interviews, if Bluecurrent is entitled to retain a law firm to enforce its controversial

---

[257] This assumes that the registration system would provide that the information would become public after a lapse of time parallel to the pre-publication period in the existing patent system.

[258] In the copyright system, developers resist registration despite rules that allow them to redact trade secrets and provide only a small sample of code. *See* Mann 1999.

[259] *See* Maurer & Scotchmer 2004 (discussing a principle of profit neutrality in enforcement of patent licenses).

networking patents, why should the right to enforce the patents be dissipated if instead it transfers the patents to a holding company so that the existence of the patents will not pollute the firm's entrepreneurial culture?

In my view, the root of the concern about trolls is not that non-operating firms hold the patents. It is simply a more fundamental dissatisfaction with the patent system itself. For example, in the case of Bluecurrent, the popular press has criticized the patents as covering technology that is so obvious as not to warrant patenting. We have an established process for resolving that question where parties disagree, and it involves litigation in courts subject to the supervision of the Federal Circuit. It may be that the Federal Circuit has made standards for obviousness unduly lax,[260] but the solution for that problem – by no means particular to the software industry – is not to bar enforcement of software patents generally. Rather, the federal courts should interpret the patent statute more coherently.[261]

Another source of concern surely is the amount of royalties awarded in suits by trolls, which in some cases have been immense sums of money. However, that concerns seems to be a byproduct of the structure of costs and benefits that I discuss above. Despite its fragmentation, the largest firms in the software industry are quite large by any standard. Thus, a reasonable royalty on any patent that one of their flagship products infringes is likely to be quite large. It may be that courts may not calculate those royalties perfectly; the imprecision of the relevant rules[262] makes it almost impossible to imagine what it would mean to calculate those royalties perfectly. Nevertheless, given the intractability of the questions, there is little reason to think that an alternate method for compensating trolls for their patents would reflect a more appropriate return on invention. Rather, it would simply substitute one imprecise scheme for another, the purpose being to lower the returns available to one particular class of patentholder.

I do not mean to suggest that the existing pattern of litigation in the software industry evinces a perfect system. I do think, however, that concerns about trolls have been overstated. Essentially, trolls are serving a function as intermediaries that specialize in litigation to exploit the value of patents that cannot be exploited effectively by those that have invented them. If that has any obvious effect, it is to increase the returns to the underlying inventors. In our patent system, that is not necessarily a bad result.

---

[260] There seems to be a consensus to that effect. *E.g.*, Eisenberg 2004 (discussing that problem).

[261] *See* Duffy 2002 (suggesting that Supreme Court review on that topic is imminent).

[262] The problem is that the royalties for an infringed patent are *not* designed to replicate the royalties that would be paid in any comparable market transaction. Accordingly, the task of the court assessing royalties is so hypothetical that neither precision nor effective appellate review can be expected. *E.g.*, Gyromat Corp. v. Champion Spark Plug Co., 735 F.2d 549 (Fed. Cir. 1984); Panduit Corp. v. Stahlin Bros. Fibre Works, Inc., 575 F.2d 1152 (6th Cir. 1978) (per Markey, C.J.). **TS** Again, to the extent there is a problem here it is not limited to the software industry or to these particular patentees in that industry.

VII. Conclusion

I do not purport to provide a definitive analysis of IP in the software industry. Indeed, I think that it is impractical to obtain the information that would be necessary to specify the precise role that IP plays in the industry. My goal is incremental. Specifically, I have attempted to accomplish two things. The first is to provide a framework to analyze the potential benefits and costs of patents in the industry. In particular, the delineation of the various indirect effects is designed to provide a basis for analysis in this and other industries. The second is to sketch the most important costs and benefits and their relations to each other. Here, the most important point surely is the differing significance of patents to firms at different stages. Within that framework, I discern a basic tradeoff, in which the main burden is the net costs of the collection of patents solely for cross-licensing, and the main benefits are the difficult to quantify benefits of patents in facilitating the formation of small firms and the licensing benefits (also of unknown size) to large firms. A full understanding of the import of that tradeoff will come only from a more developed explanation of the relative import of the contributions of large[263] and small firms to innovation.

[263] On that point, see Etro 2004 (arguing that monopolists have greater incentives to invest in R&D than outsider).

METHODOLOGICAL APPENDIX

I use interviews to collect information about the common motivations and understandings of business practices that are not readily quantifiable. There are of course a number of risks in relying on interviews. For example, there is the possibility that bias by the interviewer will taint the results of the interviews.[264] That possibility is particularly important in this type of unfocused research, because the interview scripts are not standardized.[265] There also is a significant risk that the sample of interview subjects will be biased in a way that reduces the accuracy of the information discovered in the interviews.[266] As discussed below, I have done what seems practicable to minimize those risks. In my view, however, the richness of the information available from this method far outweighs the methodological concerns.[267] The appropriate response is to proceed with caution in making firm empirical conclusions from the interviews.

The interviews typically are about 30-45 minutes long. I conducted all of the interviews personally. As is typical in this kind of research, the interviews were open-ended, without specifically scripted questions.[268] When possible, I conducted the interviews in person, but many of them were conducted by telephone. When it was possible and acceptable to the subject, I recorded the interview. If that was not acceptable, I took notes during the interview. Subject to confidentiality constraints necessary to obtain the interviews, the interview transcripts will be available on my Web site shortly after publication of this article. All of the subjects are identified in the opening footnote of the article except where the subjects requested anonymity. The transcripts include details about the positions that the subjects hold in the companies at which they are employed.

Because my goal is to understand how intellectual property affects financing practices in the industry, I attempted to speak to people who invest in startup companies – venture capitalists, angels, and banks. I also attempted to speak to people at software companies about their experiences in obtaining funding. I also spoke to people at large software companies to understand the role of IP in their assessment of potential firms for acquisition and about the role it plays in funding R&D in their own companies.[269] I also attempted to diversify geographically the interviews by contacting potential interview subjects in several of the states with large groups

---

[264] *See* SEIDMAN 1998:69, 74 (discussing that problem).

[265] *See* SEIDMAN 1998:76-77 (discussing why interviewing scripts are inappropriate in this type of research); *see also* Bertaux (1981):38-39 (discussing the need for interview "scripts" to "be modified from one interview to the next * * * according to the progress made in the understanding of [the topic]").

[266] *See* SEIDMAN 1998:44-47 (discussing that problem).

[267] *See* SEIDMAN 1998:5 (discussing the benefits of interviews to collect qualitative information).

[268] *See* SEIDMAN 1998:69-70, 76-77.

[269] IBM and Microsoft ranked first and second in the most recent release of the Software 500 by Software Magazine, available at www.softwaremag.com.

of software companies and venture capitalists (California, Massachusetts, Texas, Washington, and Michigan).[270]

The interview subjects were collected using the "snowball" method.[271] As is typical, I first used any available contacts in the industry from previous research, various institutional affiliations, and personal connections.[272] I also read widely in relevant news sources and contacted a large number of investors and developers "cold" based on news stories about recent fundings in the industry. As I interviewed subjects, I also asked for references to other potential subjects that might be willing to speak to me. As is typical for my work of this sort, I was successful in getting interviews from about one out of every four people that I contacted. I discerned no particular pattern in the likelihood that any particular person would agree to an interview.

[270] About 60% of software firms that first received venture financing in 1998 and 1999 are located in those five states.

[271] *See* SEIDMAN 1998:47 (discussing that method).

[272] This methodology is common in these types of inquiries. For example, see Hindle & Rushworth 2001. For examples of a similar methodology by other legal academics, see Baker 2001; Bernstein 2001; Bernstein 1996; Bernstein (1992); Black & Coffee 1994, Schlanger 2003.

BIBLIOGRAPHY

Bruce Abramson, *Promoting Innovation in the Software Industry: A First Principles Approach to Intellectual Property Reform*, 8 B.U. J. SCI. & TECH. L. 75 (2002)

ROB ADAMS, A GOOD HARD KICK IN THE ASS: BASIC TRAINING FOR ENTREPRENEURS (2002)

John R. Allison & Mark A. Lemley, *Who's Patenting What? An Empirical Exploration of Patent Prosecution*, 53 VAND. L. REV. 2099 (2000)

John R. Allison, Mark A. Lemley, Kimberly A. Moore & R. Derek Trunkey, *Valuable Patents* (unpublished 2003 manuscript)

ASHISH ARORA, ANDREA FOSFURI & ALFONSO GAMBARDELLA, MARKETS FOR TECHNOLOGY: THE ECONOMICS OF INNOVATION AND CORPORATE STRATEGY (2001)

Ashish Arora & Robert P. Merges, *Specialized Supply Firs, Property Rights and Firm Boundaries*, 13 INDUSTRIAL CHANGE 451 (2004)

Jonathan M. Barnett, *Private Protection of Patentable Goods*, 25 CARDOZO L. REV. 1251 (2004)

John R. Barton, *Reforming the Patent System*, 287 SCIENCE 1933 (2000)

Ann Bartow, *Separating Marketing Innovation from Actual Invention: A Proposal for a New, Improved, Lighter, and Better-Tasting Form of Patent Protection*, 4 J. SMALL & EMERGING BUS. L. 1 (2000)

Yoram Barzel, *Optimal Timing of Innovations*, 50 REV. ECON. & STAT. 348 (1968)

Daniel Bertaux, *From the Life-History Approach to the Transformation of Sociological Practice* in BIOGRAPHY AND SOCIETY: THE LIFE HISTORY APPROACH IN THE SOCIAL SCIENCES 29 (D. Bertaux ed. 1981).

James Bessen, *Patent Thickets: Strategic Patenting of Complex Technologies*

James Bessen & Robert Hunt, *An Empirical Look at Software Patents*

James Bessen & Eric Maskin, *Sequential Innovation, Patents, and Imitation* (July 2002 draft)

Bleistein, *The German Law on "Gebrauchsmuster,"* 19 J. PAT. OFF. SOC'Y 126 (1937).

BRIDGING THE ENTREPRENEURIAL FINANCING GAP: LINKING REGULATORY GOVERNANCE WITH REGULATORY POLICY (Michael J. Whincop ed. 2001)

Matthew Broersma, *Red Hat Defends Controversial Patent Applications*, ZDNET, 31 May 2002, *available at* www.news.zdnet.co.uk/business/0,39020645,2111257,00.htm

James M. Buchanan & Yong J. Yoon, *Symmetric Tragedies: Commons and Anticommons*, 43 J.L. & ECON. 1 (2000)

Dan L. Burk & Mark A. Lemley, *Policy Levers in Patent Law*, 89 VA. L. REV. 1575 (2003)

MARTIN CAMPBELL-KELLY, FROM AIRLINE RESERVATIONS TO SONIC THE HEDGEHOG: A HISTORY OF THE SOFTWARE INDUSTRY (2003)

MARTIN CAMPBELL-KELLY & WILLIAM ASPRAY, COMPUTER: A HISTORY OF THE INFORMATION MACHINE (1996)

ALFRED D. CHANDLER, JR., INVENTING THE ELECTRONIC CENTURY (2001)

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

Wesley M. Cohen, Richard R. Nelson & John P. Walsh, *Protecting Their Intellectual Assets: Appropriability Conditions and Why U.S. Manufacturing Firms Patent (Or Not)* (NBER Working Paper Series, Working Paper 7552) (Febr. 2000) *available at* www.nber.org/papers

HERNANDO DE SOTO, THE MYSTERY OF CAPITAL: WHY CAPITALISM TRIUMPHS IN THE WEST AND FAILS EVERYWHERE ELSE (2000)

Rebecca S. Eisenberg, *Consulting PHOSITA* (unpublished 2004 manuscript)

Federico Etro, *Innovation by Leaders*, 114 ECON. J. 281 (2004)

FEDERAL TRADE COMMISSION, TO PROMOTE INNOVATION: THE PROPER BALANCE OF COMPETITION AND PATENT LAW AND POLICY (Oct. 2003)

Christopher R. Fine & Donald C. Palmer, *Patents on Wall Street: Investment Banking Meets Intellectual Property*, in FROM IDEAS TO ASSETS 511

MARTIN FINK, THE BUSINESS AND ECONOMICS OF LINUX AND OPEN SOURCE (2002)

Stephen P. Fox & Guy J. Kelley, *Making Innovation Pay: Aligning Patent Rights with Business Strategy* in FROM IDEAS TO ASSETS 191

FROM IDEAS TO ASSETS: INVESTING WISELY IN INTELLECTUAL PROPERTY (Bruce Berman ed. 2002)

Joshua Gans, David Hsu & Scott Stern, *When Does Start-Up Innovation Spur the Gale of Creative Destruction*, 2002 RAND J. ECON. 571

LOUIS V. GERSTNER, JR., WHO SAYS ELEPHANTS CAN'T DANCE: INSIDE IBM'S HISTORIC TURNAROUND (2002)

Ron Gilson, *The Legal Infrastructure of High Technology Industrial Districts: Silicon Valley, Route 128, and Covenants Not to Compete*, 74 N.Y.U. L. REV. 575 (1999)

PAUL GOMPERS & JOSH LERNER, THE VENTURE CAPITAL CYCLE (2000)

PAUL A. GOMPERS & JOSH LERNER, THE MONEY OF INVENTION: HOW VENTURE CAPITAL CREATES NEW WEALTH (2001)

Stuart J.H. Graham & David C. Mowery, *Intellectual Property Protection in the U.S. Software Industry* (Feb. 2002 draft)

Jerry R. Green & Suzanne Scotchmer, *On the Division of Profit in Sequential Innovation*, 26 RAND J. ECON. 20 (1995)

Richard S. Gruner, *Better Living Through Software: Promoting Information Processing Advances Through Patent Incentives*, 74 ST. JOHN'S L. REV. 977 (2000)

Robert W. Hahn and Scott Wallsten, *A Review of Bessen and Hunt's Analysis of Software Patents* (Nov. 2003 draft available at www.ssrn.com)

Bronwyn H. Hall, *Exploring the Patent Explosion* (unpublished 2004 manuscript)

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)

Mark Haller, Edward Gold & Brian Blonder, *Avoiding Transaction Peril:Value-Based IP Due Diligence*, in FROM IDEAS TO ASSETS 37

Paul J. Heald, *A Transaction Costs Theory of Patent Law* (unpublished 2004 manuscript)

Michael A. Heller & Rebecca S. Eisenberg, *Can Patents Deter Innovation? The Anticommons in Biomedical Research*, 280 SCIENCE 698 (1998)

Thomas Hellman and Manju Puri, *The Interaction Between Product Market and Financing Strategy: The Role of Venture Capital*, 13 REV. FIN. STUDIES 959 (2000)

IDC & the Business Software Alliance, *Expanding Global Economies: The Benefits of Reducing Software Piracy* (Apr. 2, 2003)

Dennis S. Karjala, *A Coherent Theory for the Copyright Protection of Computer Software and Recent Judicial Interpretations*, 66 U. CINCINNATI L. REV. 53 (1997)

Don E. Kash & William Kingston, *Patents in a World of Complex Technologies*, 28 SCI. & PUB. POL'Y 11 (2001)

Michael Klausner & Kate Litvak, *What Economists Have Taught Us About Venture Capital Contracting*, in BRIDGING THE ENTREPRENEURIAL FINANCING GAP at 54

Samuel Kortum & Josh Lerner, *Stronger Protection or Technological Revolution: What Is Behind the Recent Surge in Patenting?*

Mark A. Lemley, *The Economics of Improvement in Intellectual Property Law*, 75 TEXAS L. REV. 989 (1997)

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

DOROTHY LEONARD, WELLSPRINGS OF KNOWLEDGE (1995)

LAWRENCE LESSIG, THE FUTURE OF IDEAS: THE FATE OF THE COMMONS IN A CONNECTED WORLD (2001)

Richard C. Levin, Alvin K. Klevorick, Richard R. Nelson & Sidney G. Winter, *Appropriating the Returns from Industrial Research and Development*, 1987 BROOKINGS PAPERS ON ECON. ACTIVITY 783

STAN J. LIEBOWITZ & STEPHEN E. MARGOLIS, WINNERS, LOSERS & MICROSOFT: COMPETITION AND ANTITRUST IN HIGH TECHNOLOGY (1999)

Clarisa Long, *Patent Signals*, 69 U. CHI. L. REV. 625 (2002)

Ronald J. Mann, *The First Shall Be Last: A Contextual Argument for Abandoning Temporal Rules of Lien Priority*, 75 TEXAS L. REV. 11 (1996)

Ronald J. Mann, *Explaining the Pattern of Secured Credit*, 110 HARV. L. REV. 625 (1997)

Ronald J. Mann, *Strategy and Force in the Liquidation of Secured Debt*, 96 MICH. L. REV. 159 (1997)

Ronald J. Mann, *The Role of Secured Credit in Small-Business Lending*, 86 GEORGETOWN L.J. 1 (1997)

Ronald J. Mann, *Secured Credit and Software Financing*, 85 CORNELL L. REV. 134 (1999)

Ronald J. Mann, *Credit Cards and Debit Cards in the United States and Japan*, 55 VAND. L. REV. 1055 (2002)

RONALD J. MANN & JANE K. WINN, ELECTRONIC COMMERCE (2d ed. Forthcoming 2005)

David McGowan, *Legal Implications of Open-Source Software*, 2001 ILL. L. REV. 1

Stephen M. Maurer & Suzanne Scotchmer, *Profit Neutrality in Licensing: The Boundary Between Antitrust Law and Patent Law* (unpublished 2004 manuscript)

JOHN MAY & CAL SIMMONS, EVERY BUSINESS NEEDS AN ANGEL (2001)

Roberto Mazzoleni & Richard R. Nelson, *Economic Theories About the Benefits and Costs of Patents*, 32 J. ECON. ISSUES 1031 (1998)

Peter S. Menell, *Envisioning Copyright Law's Digital Future* (2002 draft)

Robert P. Merges, *Rent Control in the Patent District: Observations on the Grady-Alexander Thesis*, 78 VA. L. REV. 359 (1992)

Robert P. Merges, *Contracting into Liability Rules: Intellectual Property Rights and Collective Rights Organizations*, 84 CAL. L. REV. 1293 (1996)

Robert P. Merges, *A Comparative Look at Intellectual Property Rights and the Software Industry*, in THE INTERNATIONAL COMPUTER SOFTWARE INDUSTRY 272 (David C. Mowery ed. 1996).

Robert P. Merges, *The Law and Economics of Employee Inventions*, 13 HARV. J.L. & TECH. 1 (1999)

Robert P. Merges, *Institutions for Intellectual Property Transactions: The Case of Patent Pools*, in EXPANDING THE BOUNDARIES OF INTELLECTUAL PROPERTY: INNOVATION POLICY FOR THE KNOWLEDGE SOCIETY 123 (Rochelle Cooper Dreyfuss et al., eds. 2001)

Robert P. Merges, *A New Dynamism in the Public Domain*, forthcoming U. CHI. L. REV. 183 (2004)

Robert P. Merges & Richard R. Nelson, *On the Complex Economics of Patent Scope*, 90 COLUM. L. REV. 839 (1990)

Robert P. Merges & Richard R. Nelson, *On Limiting or Encouraging Rivalry in Technical Progress: The Effect of Patent Scope Decisions*, 25 J. ECON. BEHAVIOR & ORG. 1 (1994)

Robert H. Mnookin & Lewis Kornhauser, *Bargaining in the Shadow of the Law: The Case of Divorce*, 88 YALE L.J. 950 (1979)

Petra Moser, *How Do Patent Laws Influence Innovation? Evidence from Nineteenth-Century World Fairs* (NBER Working Paper 9909) (August 2003)

DAVID C. MOWERY & RICHARD R. NELSON, SOURCES OF INDUSTRIAL LEADERSHIP: STUDIES OF SEVEN INDUSTRIES (1999)

Francesco Parisi, Ben DePoorter & Norbert Schulz, *Duality in Property: Commons and Anticommons* (Univ. Va. Sch. of Law, Law & Economics Research Papers Series, Research Paper No. 0016), *available at* www.ssrn.com

Arnold Plant, *The Economic Theory Concerning Patents for Inventions*, 1 ECONOMICA 30 (1934)

Arti K. Rai & Rebecca S. Eisenberg, *Bayh-Dole Reform and the Progress of Biomedicine*, 66 J. LAW & CONTEMP. PROBS 289 (2003)

Eric S. Raymond, *The Cathedral and the Bazaar* (1998)

KEVIN G. RIVETTE & DAVID KLINE, REMBRANDTS IN THE ATTIC: UNLOCKING THE HIDDEN VALUE OF PATENTS (2000)

JEFFREY H. ROHLFS, BANDWAGON EFFECTS IN HIGH-TECHNOLOGY INDUSTRIES (2001)

VERNON W. RUTTAN, TECHNOLOGY, GROWTH AND DEVELOPMENT: AN INDUCED INNOVATION PERSPECTIVE (2001)

Pamela Samuelson, Benson *Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 EMORY L.J.1025 (1990)

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)

Pamela Samuelson & Suzanne Scotchmer, *The Law and Economics of Reverse Engineering*, 111 YALE L.J. 1575 (2002)

ANNA LEE SAXENIAN, REGIONAL ADVANTAGE: CULTURE AND COMPETITION IN SILICON VALLEY AND ROUTE 128 (1994)

ROGER E. SCHECHTER & JOHN R. THOMAS, INTELLECTUAL PROPERTY: THE LAW OF COPYRIGHTS, PATENTS AND TRADEMARKS (2003)

Margo Schlanger, *Inmate Litigation*, 116 HARV. L. REV. 1555 (2003)

Suzanne Scotchmer, *Standing on the Shoulders of Giants: Cumulative Research and the Patent Law*, 5 J. ECON. PERSP. 29 (1991)

IRVING SEIDMAN, INTERVIEWING AS QUALITATIVE RESEARCH: A GUIDE FOR RESEARCHERS IN EDUCATION AND THE SOCIAL SCIENCES (2d ed. 1998).

Carl Shapiro, *Navigating the Patent Thicket: Cross Licenses, Patent Pools, and Standard-Setting* (March 2001)

Mathias Strasser, *A New Paradigm in Intellectual Property Law? The Case Against Open Sources*, 2001 STAN. TECH. L. REV. 4

David Teece, *Profiting from Technological Innovation: Implications for Integration, Collaboration, Licensing & Public Policy*, 15 RESEARCH POLY' 285 (1986)

D.C. TOEDT, THE LAW AND BUSINESS OF COMPUTER SOFTWARE 2002

ERIC VON HIPPEL, THE SOURCES OF INNOVATION (1988)

John P. Walsh, Ashish Arora & Wesley M. Cohen, *The Patenting and Licensing of Research Tools and Biomedical Innovation* (May 5, 2002 draft)

Rosemarie Ham Ziedonis, *Don't Fence Me in: Fragmented Markets for Technology and the Patent Acquisition Strategies of Firms* (January 2004 draft)

Jonathan Zittrain, *Normative Principles for Evaluating Free and Proprietary Software*, 71 U. CHI. L. REV. 265 (2004)