July 13, 2004 Draft

THE MYTH OF THE SOFTWARE PATENT THICKET: AN EMPIRICAL INVESTIGATION OF THE RELATIONSHIP BETWEEN INTELLECTUAL PROPERTY AND INNOVATION IN SOFTWARE FIRMS

RONALD J. MANN[*]

Table of Contents

---

[*] William Stamps Farish Professor in Law, University of Texas School of Law. I thank Allison Mann for tireless inspiration, aid, and counsel, Peggy Brundage for administrative assistance of various kinds, and Abner Greene for diligent research. For comments, I thank Oren Bracha, Sam Dinkin, Dennis Karjala, Mark Lemley, Rob Merges, Jay Westbrook, and the participants at a colloquium at the University of Texas School of Law.

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 (Road 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. The project focuses on the software industry because of the importance of that industry to the modern economy, because of the importance of innovation to that industry, and because of the well-known difficulties of accommodating traditional intellectual-property regimes (patent, copyright, and trade secret) to innovation in the industry.

This paper focuses on innovation in the hundreds of small venture-backed firms that form the bulk of the population of the industry. After a brief description of the history of the industry in Part II, Part III discusses the evidence on which the paper relies: a set of about 50 interviews of industry executives – diversified geographically, by size of company, and by role in the industry (software developers, venture capitalists, lenders, etc.). Relying on those interviews, the paper provides a detailed explication of the role that intellectual property plays in the industry.

Parts IV through VI of the paper organize the information from the interviews and situate it in the extensive literatures on venture capital investing, the economics of innovation, and patents. The first substantive topic of the paper (Part IV) is the features of startup firms that attract investment by venture capitalists – generally something about the startup that suggests a "sustainable differentiation" of the firm from its competitors. The differentiation could come from any number of advantages the firm has – a first-mover advantage, special skill of its employees, a unique approach to solving a difficult problem, or, in some cases, intellectual-property protection.

The second substantive part of the paper (Part V) discusses the role of copyright. The major point of this part is that copyright protection is of little value to startup firms. Copyright protection is designed to protect expression, and not functionality. Thus, it provides little of the protection for which venture investors are looking. The basic problem is that it does not offend copyright law if a competitor observes a software product and designs a new product that includes precisely the same functionality, so long as the competitor uses none of the "expression" from the first product. Because the competitor's customers are for the most part interested in the functionality, not the expression, this is not an important constraint. On the other hand, copyright protection does provide important protections in other areas, most obviously in protecting the later-stage firm's products from piracy. Generally, this part of the paper tells a story of unsuccessful efforts to stretch the copyright regime to do something it never was intended to do.

The final substantive part of the paper (Part VI) discusses the role of patents. Because patents do protect functionality, they have at least the theoretical potential to provide the sustainable differentiation for which investors are looking. The problem, however, is that in many sectors of the software industry innovation is not of a character that a typical patent can protect a firm from competitors: often competitors would be able to design a competing product that works around a firm's patent. Thus, despite significant increases in patenting in the industry, about 80% of venture-backed software firms do not obtain patents during the early years of their existence. The question, then, is what benefits patents do provide to those firms. This part explores several benefits, including the classic benefit of excluding competitors. In this industry at least, that benefit accrues primarily to small firms, protecting them from the competitive depredations of incumbents. Incumbents, by contrast, rarely use patents to exclude smaller firms from the industry. The part also discusses a series of less conventional benefits small firms gain from software patents: as barter in cross-licensing arrangements, in signaling their technical competence to third parties, in converting tacit knowledge into a verifiable and transferable form, and in making the firm attractive to potential acquirers.

The paper closes by discussing the implications of the patent analysis for recent debates about the value of patents in the software industry. The paper starts with a discussion of theoretical literature suggesting that free availability of patented technology is important because of the software industry's reliance on cumulative patterns of innovation. It then presents evidence about existing practices in the industry suggests that technology in fact is readily available, rebutting the prominent claims of a patent "thicket" that is supposedly stifling innovation in the industry. On the contrary, I argue, to the extent patents have an important effect in the industry, it is an effect that inures primarily to the benefit of the smaller firms trying to find a foothold from which they can compete.

THE MYTH OF THE SOFTWARE PATENT THICKET: AN EMPIRICAL INVESTIGATION OF THE RELATIONSHIP BETWEEN INTELLECTUAL PROPERTY AND INNOVATION IN SOFTWARE FIRMS

I. Introduction

For a scholar interested in understanding how intellectual property (IP) works in practice, the software industry is a natural topic for examination. Most obviously, it is important because the software industry is one of the leading areas of innovation in our economy,[1] and because the United States software industry is one of the few information technology (IT) sectors that consistently has been able to run a large trading surplus with other countries.[2] In addition, as a legal matter, software is protected not only by state trade-secret rules, but also by federal copyright and patent laws. Neither of the federal IP systems is regarded as being well suited to the industry.[3] The interplay among those methods of protection makes the role of law in the industry almost uniquely interesting.[4]

At the same time, the formal role of IP in the industry is in flux. Most prominently, recent doctrinal changes in patent law have made patents on pure software inventions directly available for the first time.[5] Those changes have led to a proliferation in software patents in recent years. Although it is difficult to get precise numbers,[6] it is

---

[1] Indeed, it is fair to say that innovation in the software and related industries has driven much of the innovation in other industries during recent decades. 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] For a comprehensive explication of the problem, see Samuelson, Davis, Kapor & Reichman (1994). For more recent commentary, see Gruner (2000):992-98 (general doctrinal analysis); Haynes (1999) (trenchant argument that software protection slows innovation that would proceed at the proper pace if patent law were left unimpeded).

[4] For a coherent discussion of that interplay as a doctrinal matter, see Dennis S. Karjala, *The Relative Roles of Patent and Copyright in the Protection of Computer Programs*, 17 J. MARSHALL J. COMPUTER & INFO. L. 41 (1998).

[5] For general discussion of the rise of the software patent, see Burk & Lemley (2002):1160-73; Cantzler (2000).

[6] The problem is that the system by which the Patent and Trademark Office classifies patents does not have a separate class (or classes) into which software-related inventions fall. Thus, all data about the number of software patents to date have been produced by rough estimates of various kinds. *See* Allison & Lemley (2000): 2115 & n.51. In a forthcoming paper with John Allison, I respond to that problem by looking from the perspective of the patentee; we collect and analyze a dataset of patents granted to the 500 largest firms in the software industry.

clear that the Patent and Trademark Office is granting more than 10,000 software patents each year.[7]

As the issuance of software patents increases, scholars have expressed considerable doubt that patents are useful for the software industry *at all*. It was a prominent topic in recent FTC hearings on the balance of competition and patent law.[8] The broadest form of the criticism – associated with Carl Shapiro and Larry Lessig – argues that the rapid proliferation of software patents has created a "patent thicket" that deters innovation, particularly by small firms that are not well placed to compete against the portfolios of their larger and better-heeled competitors.[9] John Barton has made the criticism even more general: arguing 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.[10] That concern has been buttressed by a widely discussed (though not published) empirical study by James Bessen and Eric Maskin suggesting that R&D spending has been declining in the software industry at the same time that patent issuance has been rising.[11] A more guarded concern expressed by Mark Lemley in papers with Julie Cohen and Dan Burk suggests that, at a minimum, software patents should be construed to have a narrow scope.[12] The issue also is under debate in Europe, where the European Union is in the

---

[7] 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.

[8] FEDERAL TRADE COMMISSION, TO PROMOTE INNOVATION: THE PROPER BALANCE OF COMPETITION AND PATENT LAW AND POLICY (Oct. 2003) [hereinafter FTC REPORT].

[9] *See* LESSIG (2001):205-15; Shapiro (2001); FTC REPORT:52-53.

[10] Barton (2000). Doubts about whether the patent system as a whole causes an increase in innovation are not new. *See* Plant (1934):33-37. Moreover, all of these studies assume that increases in innovation are uniformly good: they 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).

[11] *See* Bessen & Maskin (2002). Bessen advances that work in a recent working paper. Bessen (2003). Whatever the theoretical implications of Bessen's claims might be if they were true, I discuss at the end of this paper evidence indicating that the factual basis for them – that R&D spending is unusually low in the software industry – is at least overstated.

[12] *See* Burk & Lemley (2003):82-90; Cohen & Lemley (2001). Lemley's work, however, does not suggest a uniform antipathy to software patents. Lemley (1997) hails the primacy of patent protection for software as supporting a beneficial trend toward reusing specific software components instead of having each developer rewrite similar pieces of code from scratch. Also, Burk & Lemley acknowledge that TRIPS would bar any special statutory treatment for software. Burk & Lemley (2003):110-11. Rather, they argue, courts should use a variety of policy levers to

midst of a process that may – or may not – result in EU-wide availability of patents for software.[13]

In the face of the controversy about software patents, it is astonishing how little work has been done directed specifically at the software industry. There has been some serious doctrinal analytical work – most notably the seminal *Manifesto* by Pam Samuelson and her co-authors and the work of Mark Lemley and his co-authors discussed above.[14] There has not been, however, any substantial published empirical work about the role of patents in the industry.[15] Aside from two studies examining the European industry,[16] the most prominent paper that provides any empirical evidence about the software industry itself is the Bessen & Maskin paper mentioned above.[17] That paper, however, ends its inquiry in 1995, only one year after the en banc decision of the Federal Circuit in *In re Alappat* that cleared the way for software patents.[18]

---

treat software patents appropriately in light of the patterns of innovation in the industry. *See* Burk & Lemley (2003):210-20.

[13] *See* EC Commission, *Proposal for a Directive of the European Parliament and of the Council on the Patentability of Computer-Implemented Inventions* (Feb. 20, 2002), *available at* http://europa.eu.int/comm/internal_market/en/indprop/comp/com02-92en.pdf (last visited Apr. 18, 2003).

[14] Samuelson, Davis, Kapor & Reichman (1994); *see also* Samuelson (1990). The *Manifesto* provides a thorough description of the software industry as it had developed through the early 1990's, together with a thoughtful proposal for a *sui generis* system of IP protection that would combine particular features of the copyright and patent systems.

[15] That is clearest by the lack of certainty about the most basic fact – how many software patents there are, *see supra* note 7 – despite the frequent academic complaints that there are far too many. While this paper was being prepared, I located a draft of Thakur (2003). That paper does analyze a dataset of patents held by software firms, but it is limited to the very largest firms in the industry and thus does not contribute much to an understanding of the role of patenting in the industry as a whole.

[16] *See Micro-and Macroeconomic Implications of the Patentability of Software* (2001); Tang, Adams & Paré (2001). Although those studies are interesting – I draw parallels to them below – they have limited value in assessing the role of IP in the United States because the European software industry is so much smaller than the American one, *see* Campbell-Kelly (2003):23, and because its structure raises such different policy concerns, see Kahin (2002):13-18.

[17] To get a sense for the problem, the large empirical study of patenting by Allison & Lemley that is the basis for Allison & Lemley (2000) includes only 76 software patents. Allison & Lemley (2000):76. Allison & Tiller also are working on a not-yet published study on business-method patents, which includes a substantial body of certain classes of software patents. Allison & Tiller (2003). Finally, Bessen also is working on a number of as-yet unpublished papers (some of which I discuss below).

[18] 33 F.3d 1526 (1994).

Moreover, like all of the studies to date, their papers focus on software patents, not the software industry. Because there is considerable ambiguity about exactly what a software patent is – they do not fall into any specific PTO class – and because many patents in the classes that indisputably do constitute software patents are held outside the industry, that work says little or nothing about the software industry itself. As I suggested above, software is unusual among patentable goods in its interaction with all sectors of the economy. Thus, there are *some* software features in a wide variety of otherwise unrelated products.

That should not, however, obscure the fact that there is a large, highly populated, highly innovative, and successful software industry – and that any positive and adverse effects of doctrinal rules related to software are likely to be concentrated in that industry. To the extent that existing papers do consider some of the firms in that industry, they tend to focus on large publicly traded firms. That emphasis causes the papers to miss the implications of various features of the software industry that are crucial to this inquiry: the immense number of firms in the industry (indicating low barriers to entry) and the variation among the sectors in the industry.

This study meets those needs with a contextual study of the role that patent and copyright law protections play in fostering innovation in the software industry. Specifically, this study attempts to understand as precisely as possible the role that intellectual property plays in generating investments in startup firms in the software industry. From a broader perspective, it is a case study attempting to provide some concrete analysis of the suggestion of Mazzoleni and Nelson that in some industries we can find value for patents entirely in their ability to induce commercialization of inventions.[19] The gist of that theory is that patents induce commercialization by helping inventors to obtain the financing that they need to commercialize their product.[20] As Mazzoleni and Nelson point out, that theory suggests that the patent process is most useful in cases in which innovation would occur in "a small firm that must marshal outside funds,"[21] exactly the situation that we see in the software industry.

This paper rests on a series of more than fifty interviews with industry executives. I included those who create the software (executives at startup companies) those who finance its creation (venture capitalists, angel investors, and banks), and those who represent them in those endeavors (lawyers in firms and also at the various companies). The wide-ranging interviews were designed to gain as much contextual information as possible about the motivations for patenting (or not patenting) in the industry. I rely on interviews because the goal of this study is not to understand the quantitative question –

---

[19] *See* Mazzoleni & Nelson (1998):1040-42 (crediting Eisenberg (1997) as the source of the theory).

[20] *See* Mazzoleni & Nelson (1998):1040-42; Eisenberg (1997).

[21] Mazzoleni & Nelson (1998):1041.

do software startups obtain patents?[22] – but rather to examine the more fundamental question discussed above: exactly how (if at all) patents contribute to innovation. Because that question turns on subjective motivations that are not readily quantifiable, in-depth interviews are the best method of obtaining information.[23]

The body of the paper has five parts. Part II provides a brief overview of some key features of the software industry: the origins of the industry, the early role of intellectual property in the industry, and the importance of small firms in industry innovation. Part III discusses the methodology for the interviews. Parts IV through VI then present the results of the interviews. First, Part IV discusses the perspectives of those who provide funds for early-stage companies: venture capitalists, bank lenders, and angel investors. The principal point of that part is that the investors are primarily concerned with sustainable differentiation: does the company have something that differentiates its product from competing products in a way that can be sustained over time. Parts V and VI build on that point to assess the extent to which copyrights and patent can assuage that concern.

Part V discusses the role of copyright. The Part proceeds in three steps. First, I explain why copyright law plays no substantial role in providing the sustainable differentiation that investors seek in deciding whether to invest in early-stage software firms. Second, I discuss the special case of those companies that lend to startups. Those lenders traditionally have taken care to obtain a perfected interest in copyrights of their borrowers. As I explain, that practice does not indicate that the copyrights are valuable for the startups. On the contrary, it indicates that in many ways the extension of copyright law to cover unregistered software is an unwanted nuisance. Third, I attempt to place the role of copyright law in a broader context of the software industry as a whole. Despite its disutility for attracting startup funding, copyright protection plays several key roles in the structure of the industry. Among other things, it is particularly important in consumer and international markets where blatant copying of object code – piracy – is a serious problem. Similarly, it also is important in protecting companies against competition by former employees or business contacts who might leave a company with substantial amounts of copyright code and attempt to reuse it in a later venture.

---

[22] A companion paper with Tom Sager analyzes quantitative evidence about possible correlations between the success of software startups in obtaining funding and the size of their patent portfolios. That study (which is discussed in this paper where relevant) indicates that only about 20% of the software startups that received venture funding in the late 1990's have received a patent. Indeed, even among software startups that received venture funding in the mid 1990's (many of whom are now public companies), only about 30% have received patents through the present date.

[23] The only similar study – the examination of the semi-conductor industry in Hall & Ziedonis (2001) – relies on a similar series of interviews in that context. *See also* Walsh, Arora & Cohen (2002) (relying on interviews in the biotechnology industry to reject the "anti-commons" thesis articulated in Heller & Eisenberg (1998)).

Part VI, about the role of patents, is the first effort in the literature to explain at a micro-level exactly how and why small firms might – or might not – benefit from obtaining patents. The basic thesis of this part is that patents have an important effect on the structure of the industry by making it easier for smaller startup firms to obtain financing and then to protect themselves from incumbent competitors long enough to gain significant market traction. More directly, I argue that patents are much more likely to aid the creation and survival of small firms than they are to foster the continued dominance of existing firms. The analysis proceeds in two steps. First, relying largely on the interviews, I explain the specific benefits that patents can provide to the firms that obtain them. At the core of those explanations is the basic potential for patents to secure for the smallest of firms the ability to practice its innovation in some narrow area protected by the patent. That potential might differ greatly from sector to sector and patent to patent, but it does seem to play an important structural role in the industry.

Nonstructural explanations also appear. First, patents are said to be useful as "barter" in cross-licensing agreements that the firm would enter into if it reached a sufficiently mature stage to be a significant player in the industry. Second, patents are said to provide a "signal" of engineering discipline and market understanding: 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. Third, patents are said to be important for "marketing": convincing the investors in public markets that the company's technology is valuable. The idea here 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, customer's needs for that product, and the personnel that the company employs to execute its business plan. Later-stage investors, by contrast, are said to be less willing to undertake such careful evaluations and rely (less thoughtfully) on the mere existence of patents in the company's portfolio.

Part VI then considers the policy implications of this study. Although there is much work that remains to be done,[24] the results of this initial study are relevant to the concerns in the existing literature about the role of patents. That literature has recognized the importance of industry differences along two separate dimensions: innovation proceeds differently in some industries than it does in others; and patents work differently in some industries than they do in others. With respect to the software industry in particular, several scholars[25] have suggested that patents do not contribute to innovation in the industry at all. Although my evidence is inherently anecdotal – and thus not definitive – it does leave me strongly inclined to reject that argument. I make two main points. First, the critics generally have failed to appreciate the variety of ways in which

---

[24] In particular, a comprehensive understanding of software patents in the industry requires work about the role that software patents play in large firms. As mentioned above, I am in the midst of work with John Allison on a study of that topic.

[25] Bartow (2000) (focusing on software); see Kash & Kingston (2001) (arguing that patents do not work in complex industries because they are used as bargaining chips).

software patents can contribute to innovation. There are several contributions software patents make, some of which have the potential to provide a substantial net benefit for the industry. Second, most of the criticisms rest on the implicit or explicit assumption that the spread of patents through the industry will hinder innovation because the patented technology in fact is not available to developers at smaller and less well-established firms. Although the evidence discussed here cannot justify a categorical rejection of those claims, it does make it difficult to take those concerns seriously. Taken together, the volume of multiple-round financing of patentless startup firms, as well as the cross licensing at low transaction costs that pervades the industry, suggest that the existing body of patents are not stifling innovation in startup firms in any substantial way. Indeed, it strongly suggests that patents are central to the creation of the highly fractionated and innovative industry that has developed in the last decade.

II. The Software Industry

To situate the argument I present about the relation between IP and the structure of the software industry, it is important to start with some basic introductory facts about the software industry. The software industry generally is regarded as originating in the mid-1960's.[26] 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[27] and the shortage of the labor needed for each firm to make its own software.[28] The single most crucial event was the decision of IBM in late 1968 to "unbundle" its software from its hardware.[29] From that point, 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.[30]

The industry was not, however, fated to retain the more or less 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 the succession of a series of events that have repeatedly remade the terrain of the industry. The first of these was the introduction of the personal computer in the mid-1970's, which led to a largely separate set of companies

---

[26] The discussion of the early period draws heavily on CAMPBELL-KELLY (2003):89-119.

[27] 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.

[28] 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.

[29] 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.

[30] *See* CAMPBELL-KELLY (2003):121-73.

producing software for personal computers.[31] 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.[32] Yet another sector of the industry that arose by that time is the massive sector producing games and other entertainment software.[33] Finally, the rise of the Internet has brought first a tremendous influx of capital into the industry and then a subsequent crash and weeding out when companies were not able to produce results sufficient to justify the elevated equity valuations of 2000 and 2001.

The most startling thing about the industry as it has matured is the astonishing lack of concentration in the industry – a facet that has considerable implications for the competitive structure of the industry and its openness to innovation.[34] 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.[35] More than 500 firms in the industry had a million or more dollars in sales in 2002, even after the cutbacks and contractions in the industry at the turn of the millennium.[36] In that same year, 209 new 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 for the industry.[37] 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 terms of revenue had less than thirty percent of the revenues of the industry as a whole – an astonishingly small figure when we consider the major industries in which there are not ten players in the entire country.

---

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

[32] *See* CAMPBELL-KELLY (2003):231-66.

[33] *See* CAMPBELL-KELLY (2003):269-301.

[34] This is nothing new. *See,e.g.* , CAMPBELL-KELLY (2003):167 (noting that the software industry even by the early 1980's was much less concentrated than the parallel hardware industry out of which it had grown).

[35] I am aggregating 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).

[36] The smallest firm in the 2002 *Software 500* was PaperClip Software, Inc., with sales of $1.2 million.

[37] Overall 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.

The lack of concentration of the industry is underscored by the immense variety of products, which produces such a large number of market niches that no single company (even Microsoft or IBM) could maintain market power in all of them.[38] The most recent promulgation of the Software 500,[39] for example, describes the major sectors with the following lengthy list (with characteristic firms in parentheses): operating systems (Microsoft), middleware (IBM), database (Oracle), storage management (Unisys), financial applications (Intuit), computer-aided design (Autodesk), e-business applications (SAP and Software AG), application development tools (Sun and Compuware), infrastructure (Computer Associates), customer-relationship management (Siebel Systems), enterprise-resource planning (PeopleSoft), security (Symantec, Verisign), supply chain (Aspect Technology), business intelligence (Cognos), content/document management (Vignette), vertical industry applications (Per-Se Technologies), wireless (Qualcomm), and IT services (Hewlett-Packard).

The industry also has recorded an astonishing 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).[40] During that period, the wages earned by employees grew at an average annual rate of 7.8 percent, for a 2000 average wage of $80,900, the highest in any of the information technology-producing industries.[41]

For present purposes, what is most important about the track of development 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 from time to time, it is fair to say that "[t]he United States has traditionally embraced strong protection for computer software."[42] In the early days, it was generally believed that it was "trivially easy to replicate" the software program of a competitor.[43] When initial efforts by major industry players to obtain patents on their products were unsuccessful,[44] firms (and Congress) turned to copyright as an alternative.[45] The Copyright Office formally decided to permit

---

[38] *See* CAMPBELL-KELLY (2003):231-66 (chapter headed "Not Only Microsoft" discussing various successful strategies for competing with Microsoft).

[39] The unusual fractionation of the industry is underscored by the very existence of a Software 500 – what other single industry has such a thing?

[40] UNITED STATES DEPARTMENT OF COMMERCE, ECONOMICS AND STATISTICS ADMINISTRATION, DIGITAL ECONOMY 2002, p. 43.

[41] *Id.*

[42] Merges (1996a):277.

[43] CAMPBELL-KELLY (2003):107.

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

[45] 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).

registration of programs in the mid-1960's.[46] 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.[47] Thus, until the late 1980's copyright provided relatively strong protection for software.[48]

However, as the courts began to face more of these cases, the courts eventually narrowed copyright protection so that it ceased to provide robust protection.[49] The key problem that the courts increasing faced 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."[50] 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 entire programs. The court limited protection to specific pieces of the program limited to "expression."[51] Two years later, the Ninth Circuit refused to protect Apple's graphical user interface from appropriation by Microsoft.[52] The façade of pervasive copyright protection came crashing to a definitive ruin with the celebrated decision of the First Circuit three years later in *Lotus Development Corp v. Borland Int'l, Inc.*[53]

Yet, long before those decisions limited the overarching importance of copyright protection, major firms in the industry already had begun turning to patent protection. Direct protection of software patents was difficult in the wake of the Supreme Court's 1972 decision in *Gottschalk v. Benson*.[54] Still, several of my interviews suggest that software patents were easy to obtain. Because much of the software through the 1980's was being produced by hardware firms, patents easily could be obtained on an object (a microprocessor), programmed to accomplish the relevant function.[55] To be sure, that artifice was not effective in the 1990's when large software-only firms like Microsoft started to play major roles, because firms like Microsoft could appropriate the innovation of those patents in a software program without infringement. By that time, however,

---

[46] Menell (2002):16.

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

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

[49] Graham & Mowery (2002):8-9.

[50] Samuelson et al. (1994):2351.

[51] 23 U.S.P.Q.2d 1241; Merges (1996a):277-278.

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

[53] 49 F.3d 807 (1st Cir. 1995) (no copyright protection for pull-down menus in spreadsheet program).

[54] 409 U.S. 63 (1972) (barring patent on algorithm); *see* Menell (2002):16.

[55] Witek; Hill.

patent doctrine had begun to loosen so that patent protection was available, at least to those firms that were sufficiently familiar with the process to exploit it.[56]

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. While copyright protection seemed likely to provide adequate protection, many who were active in the industry thought that patent protection would be counterproductive.[57] 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.[58] The question that this article addresses is how those forms of protection have developed in practice as the industry has matured. The remainder of this paper investigates those questions.

III. Interview Methodology

As in much of my past work, I use interviews to collect information about the common motivations and understandings of business practices that are not readily quantifiable.[59] 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.[60] That possibility is particularly important in this type of unfocused research, because the interview scripts are not standardized.[61] 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.[62] 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.[63] The appropriate response is to proceed with caution in making firm empirical conclusions from the interviews.

---

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

[57] LESSIG (2001):207-08; Merges (2002):15; Graham & Mowery (2002):10-11.

[58] A companion paper with John Allison analyzes the patents of the leading 500 software firms.

[59] Mann (1996); Mann (1997); Mann (1997a), Mann (1999), Mann (2002).

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

[61] *See* SEIDMAN (1998): 76-77 (discussing why interviewing scripts are inappropriate in this type of research); *see also* 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, 38-39 (D. Bertaux ed. Int'l Sociological Ass'n 1981) (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]").

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

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

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.[64] 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.[65] I also attempted to diversify geographically the interviews by contacting potential interview subjects in several of the states with large groups of software companies and venture capitalists (California, Massachusetts, Texas, Washington, and Michigan).[66]

The interview subjects were collected using the "snowball" method.[67] As is typical, I first used any available contacts in the industry from previous research, various institutional affiliations, and personal connections.[68] 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.

---

[64] *See* SEIDMAN (1998):69-70, 76-77.

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

[66] Data I have collected for a related project indicates that about 60% of software firms that first received venture financing in 1998 and 1999 are located in those five states.

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

[68] 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).

IV. VC Investment

I turn now to the results of the interviews, the subject of the remaining three parts of the paper. Because my focus in this project is on understanding the ways in which IP protection might facilitate funding of software startups, I begin in this part with a general discussion of the nature and purposes of venture-capital investment. The next part discusses copyright protection. The final part discusses patent protection.

The development of cutting-edge software is an expensive and time-consuming task. 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 time and effort. That time and 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.[69] At that point, in most cases, the firm will turn to institutional investment.

The most common[70] source of that investment is a venture capitalist.[71] Generally, 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.[72] 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.[73] For the

---

[69] *See* MAY & SIMMONS (2001):32-33 (discussing the use of "friends-and-family" money before turning to institutional financing).

[70] Angel investors provide a source of financing that is parallel (or, in some cases, preliminary) to venture-capital investment. Their role in particularly 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. For a general discussion, see MAY & SIMMONS. 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 that the text discusses in detail. *See* MAY & SIMMONS (2001):170-71 (reporting advice from Guy Kawasaki); Jackson Interview; Lord Interview.

[71] 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.

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

[73] For a wide-ranging collection of quantitative analysis of those topics, see GOMPERS & LERNER (2000). For a good summary, see Klausner & Litvak (2001).

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. The premise of intellectual property protection 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 reason why the monopoly could support such a flow of investment is that the monopoly – at least in theory – could create 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.[74] Rather, the typical investor is likely to start by focusing on basic issues that validate the core competency of the firm to execute its concept successfully.[75] For example, investors will be interested in such things as expertise in the relevant market and the skills of the management team.[76] One remarked: "Every company of mine that has failed has been mismanagement of executives, not technical failure."[77] Similarly, even before investors consider whether a firm can protect a position as a market leader, they will want to know whether their product is one that customers need so desperately that the firm can gain a significant place in the market.[78]

Still, for firms that have a credible product idea and the expertise to implement it, venture capitalists plainly accept the basic idea that their goal is to identify firms that will have sufficient market power to earn extraordinary profits. Intellectual property protection is important only indirectly, as a tool that sometimes can provide that market power. The key is, in the words of one of my most lucid interview subjects, "sustainable

---

[74] A typical comment is that entrepreneurs are naïve if they think they have discovered a valuable product idea for which there is no competition: "This is complete and utter BS. Every product has competition, even if it's just 'I want to keep my money.'" 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 (arguing that "the existence of competition * * * functions as its own form of market validation"); MAY & SIMMONS (2001):170.

[75] *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].").

[76] *See* ADAMS (2002):27-39 (discussing "execution intelligence"), 125-52 (discussing the importance of the management team in securing funding); MAY & SIMMONS, *supra* note 69, at 171.

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

[78] *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.

differentiation"[79]: something special about the particular firm that will enable it to do something that its competitors will not be able to do for the foreseeable future. The interviews also reflect more picturesque terminology – referring to "secret sauce"[80] or "magic dust."[81] 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.[82]

For example, it was common for investors to refer to lead-time or first-mover advantages.[83] The premise of those discussions 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. Interestingly enough, the expectation generally rested on the perception that a firm could maintain a lead on its rivals as long as it kept improving its technology as quickly as its competitors.[84] I rarely if ever heard investors (as opposed to developers) who expected portfolio firms to obtain and retain a strong market position through "lock-in" or "bandwagon" effects.[85]

---

[79] Lee Interview:1.

[80] Lee Interview:1.

[81] *See* Weghorst Interview:3.

[82] *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.").

[83] *See* ADAMS (2002):73-94 (discussing importance of getting to the market quickly); 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.

[84] *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.}

[85] "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).

That is not to say that intellectual property protection is not important. It is clear, however, that different investors have different views about it.[86] Some feel that intellectual property always is important, and claim that they never invest without strong patentable technology.[87] However, even those investors often 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.[88] Others, however, particularly those that emphasize early-stage companies, say that IP protection is generally unimportant for software investments.[89] Still others take a middle position, holding that IP protection matters some, but not all, of the time.[90] Most of those who addressed the subject recognized differing perspectives on the point and argued that those with differing perspectives are misguided.[91] The most likely explanation is that the different investors are simply implementing different investment models based on their particular expertise.[92]

V. The Role of Copyright

The software industry presents a puzzle for the law of copyright. On the one hand, the purpose of copyright law is to protect expression, as opposed to ideas or functionality.[93] Indeed, the Supreme Court has gone so far as to suggest that the Constitution could forbid an extension of copyright law beyond the protection of

---

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

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

[88] *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).

[89] *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.").

[90] Stephenson Interview:1.

[91] *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).

[92] I hope to study that question in a forthcoming paper, analyzing correlations between patenting practices of portfolio companies and the identity of the venture capitalists that invest.

[93] *Baker v. Selden*, 101 U.S. 99, 103-05 (1879); Copyright Act § 102(b) (excluding from copyright protection "any idea * * * process, system, [or] method of operation * * * embodied in [a copyrightable] work").

expression.[94] On the other hand, the purpose of software products is to enhance the functionality of the computers (or other objects) that the software operates.[95] How, then, can copyright function as an integral part of the system for providing intellectual-property protection for software?[96]

One answer is that it is a historical accident. Within a few years of the birth of the software industry in the 1960's, the question of how software should be protected became bound up with the lengthy project to reform the Copyright Act of 1909. Shortly after that project finally resulted in the Copyright Act of 1976[97] – the basis for copyright law to this day – software was firmly entrenched in the statute: Section 101 specifically defines a "computer program" and the statute includes a variety of rules designed to bring software into the statute.[98] Yet, what Congress did not do was reconcile the fundamental tension in using a statute consciously limited to protection of expression to bolster an industry dedicated to enhancing the utility of computers.

Because of the importance of the software industry in the United States economy, it is not surprising that the courts eventually had to grapple with this tension. As discussed above, a series of cases in the early 1990's struggled to distinguish between the expressive and functional elements of computer programs, granting protection to the former but denying protection to the latter.[99] Similarly, courts have struggled with cases

---

[94] Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 348 (1991) (discussing originality requirement).

[95] *See* Bruce Abramson, *Promoting Innovation in the Software Industry: A First Principles Approach to Intellectual Property Reform*, 8 B.U. J. SCI. & TECH. L. 75, 122-23 (2002).

[96] For previous notices of this puzzle, see, *e.g.*, Abramson, *supra* note 95, at 77-78, 122-23; Dennis S. Karjala, *A Coherent Theory for the Copyright Protection of Computer Software and Recent Judicial Interpretations*, 66 U. CINCINNATI L. REV. 53, 53-54 (1997); Pamela Samuelson, Randall Davis, Mitchell D. Kapor & J.H. Reichman, *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2350 (1994) ("Copyright law is mismatched to software, in part, because it does not focus on the principal source of value in a program (its useful behavior).").

[97] For a comprehensive and critical discussion of the development of that Act, see Jessica D. Litman, *Copyright, Compromise, and Legislative History*, 72 CORNELL L. REV. 857 (1987).

[98] *See* Karjala, *supra* note 96, at 72-77 (discussing those rules).

[99] *E.g.*, Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222 (3rd Cir. 1986) (copyright protection for most aspects of software program); Computer Assoc. Int'l v. Altai, 982 F.2d 693 (2nd Cir. 1992) (more limited protection through an abstraction-filtration-comparison test); Autoskill, Inc. v. Nat'l Edu. Support Sys., Inc., 994 F.2d 1476, 1499 (10th Cir. 1993) (copyright protection for software keying procedure); Lotus Dev. Corp. v. Borland Int'l, 49 F.3d 807, 815 (1st Cir. 1995), *aff'd by equally divided Court*, 516 U.S. 233 (1996) (no copyright protection for keystrokes necessary to operate Lotus spreadsheet). For discussion and categorization of such cases, see Abramson, *supra* note 95, at 123-24, 131-32; Karjala, *supra* note 96, at 78-112; Menell (2002):20-29.

involving reverse engineering. Those cases generally justify rules that permit parties that are engaged in functional development to use expressive aspects of the programs in ways that otherwise would be prohibited.[100] More abstractly, leading academics have struggled to develop coherent frameworks within which to provide appropriate IP protection for software innovation. Some – most notably Pamela Samuelson and her co-authors – have suggested entirely new bodies of law.[101] Others, the most successful of whom probably is Dennis Karjala, have worked to find coherence in the overlapping protections that copyright and patent afford.[102]

Stepping back from that doctrinal and scholarly morass, the key question in evaluating this body of law should be how well its extension to the software industry serves its fundamental purposes.[103] After a quarter of a century, enough time has elapsed to examine that question empirically. Starting from the premise that the fundamental purpose of the intellectual-property system is to stimulate production in the covered sectors of the economy,[104] the basic question for me is how – and whether – copyright law facilitates the flow of money to enterprises engaged in the business of software innovation.

That is not a question readily susceptible of quantification. Unlike patent protection, copyright protection is not dependent on a decision to register an innovation with a government office; the protection attaches when the software is written.[105]

---

[100] The leading case is Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1527 (9th Cir. 1992) (allowing Accolade to reverse engineer Sega's programs to create Sega-compatible games). For discussion, see Julie E. Cohen, *Reverse Engineering and the Rise of Electronic Vigilantism: Intellectual Property Implications of 'Lock-out' Programs*, 68 S. CAL. L. REV. 1091 (1995); Karjala, *supra* note 96, at 112-16; Menell (2002):29-46; David A. Rice, Sega *and Beyond: A Beacon for Fair Use Analysis . . . At Least As Far As It Goes*, 19 U. DAYTON L. REV. 1131 (1994).

[101] Samuelson et al., *supra* note 96, at 2378-420.

[102] *See* Karjala, *supra* note 96; Menell (2002) (arguing that the "thin" protection copyright provides for software does not interfere with the protection afforded by patent law).

[103] There is comparatively little scholarship on how well copyright functions even in the core literary industries for which it was developed. For one significant example of an effort to examine that question, see Stephen Breyer, *The Uneasy Case for Copyright: A Study of Copyright in Books, Photocopies, and Computer Programs*, 84 HARV. L. REV. 281 (1970).

[104] *E.g.*, Karjala, *supra* note 96, at 53; *cf*. U.S. CONST. art. I, § 8, cl. 8 (authorizing copyright protection "To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries").

[105] Graham & Mowery (2002) analyze data about copyright registrations by large packaged software firms, which indicate that large firms are registering their software copyrights less frequently than they once did. It is difficult to draw conclusions from that data given the limited connection between registration and copyright protection and the obvious reasons why parties might not register software even when they wished to rely on copyright protection. Mann (1999).

Accordingly, the only practicable way to understand the significance that copyright has for the industry is to understand the role it plays in the business decisions of those in the industry. The basic problem is to determine whether the remnants of copyright protection for software play any significant role in the modern software industry. This part of the paper assesses that question in three steps. First, it explains why copyright cannot provide the sustainable differentiation that investors seek when they assess potential investments. Second, it discusses the special case of lenders, who plainly *are* interested in copyright protection. Third, it discusses the role of copyright in later stage software firms.

A. 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. On that specific question, the answer is a resounding "no." The basic problem with copyright protection for software is that the legal system for copyright is designed *not* to protect functionality. Because functionality in most cases is the aspect of software products that makes them attractive to customers,[106] 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."[107] Another argued: "I've been in the software

---

[106] 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) than those for software programs. Knowledgeable industry executives recognize this distinction as crucial in the negotiation of contracts related to the exploitation of those works. *See* Interview with Jeffrey Koontz, Senior Attorney, Consumer Group, Microsoft (Feb. 4, 2003, Redmond, WA) [hereinafter Koontz Interview] [transcript on file with author] (transcript at 1).

[107] Harding Interview, *supra* note 75 (transcript at 1); *see* Telephone Interview with Rob Beauchamp, Chief Technology Officer, Journee (Apr. 3, 2003) [hereinafter Beauchamp Interview] [transcript on file with author] (transcript at 8) ("To what extent does [copyright] keep people from stealing your ideas or your product? None."); Telephone Interview with Tim Costello, President, Builder Homesite (Nov. 26, 2002) [hereinafter Costello Interview] [transcript on file with author] (transcript at 1) ("Generally, I think that most people in the software industry don't think it is worth all that much."); Telephone Interview with Ken Kalinoski, Chief Technology Officer, Forgent (Feb. 19, 2003) [hereinafter Kalinoski Interview] [transcript on file with author] (transcript at 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."); Telephone Interview with Anne Kelley, Senior Attorney,

business for 20 years. Copyrights are worthless. They are totally worthless."[108] 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. Patents are inter-industry mechanisms for creating value. Copyright is creating protection between the industry and the channel or end customers.[109]

Thus, a common theme was that copyright was not just of small value, but completely unknown to executives at startups. For example, one experienced lawyer opined that "[f]or startup companies, the Copyright Office is a foreign place."[110]

The most obvious problem with that argument relates to reverse engineering.[111] Generally, the expression that copyright protects in computer software is in the lines of

---

Patents, Microsoft (Feb. 27, 2003) [hereinafter Kelley Interview] [transcript on file with author] (transcript at 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, *supra* note 83 (transcript at 2) ("We don't make copyright a big issue."); Lee Interview, *supra* note 77 (transcript at 1) (copyright is "not a focus area for us" because it is "not useful"); Murphree Interview, *supra* note 87 (transcript at 1).

[108] Adams Interview, *supra* note 89 (transcript at 1).

[109] Interview with David Kaefer, Director, Corporate Initiatives, Microsoft (Feb. 4, 2003, Redmond, WA) [transcript on file with author] (transcript at 2). For a similar emphasis on the vertical – rather than horizontal – value of copyright protection, see D'Eath Interview, *supra* note 83 (transcript at 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(C)(1)), copyright's role in preventing piracy is arguably its most important role in the software context.

[110] Interview with Edward A. Cavazos, Partner, Andrews & Kurth, LLP (Sept. 17, 2002, Austin, TX) (transcript at 1) [transcript on file with author]; *see* Telephone Interview with Michael Abbott, President, Composite Software (Mar. 21, 2003) [hereinafter Abbott Interview] [transcript on file with author] (transcript at 8) ("[The costs [for copyright protection] are not too significant but I'm not as knowledgeable frankly, in that area, as some of the patents."); D'Eath Interview, *supra* note 83 (transcript at 6) (referring to registration requirements, complained that copyright "to me is more of an annoyance than it is a real protection"); Weghorst Interview, *supra* note 81 (transcript at 1) ("In the early days of software start-ups * * * copyright protection is not even on the radar screen in terms of importance.").

[111] 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 114 (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.

code of which the program consists.[112] Thus, although copyright does not prohibit a competitor from writing a completely new program that includes the functionality of the existing program, it does generally bar a competitor from taking the existing code to produce that program.[113] 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 enough, 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."[114] As one venture investor explained: "The difficulty normally is managing the people, not solving the problem. The code won't look the same, but the functionality will."[115] One developer explained that the difficulty in coming up with a successful

---

[112] *See, e.g.*, Karjala, *supra* note 96, at 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, *supra* note 96, at 75-77, 94-112 (arguing that copyright protection does not extend to software interfaces).

[113] *See, e.g.*, Karjala, *supra* note 96, at 72-75.

[114] Weghorst Interview, *supra* note 81 (transcript at 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 et al., *supra* note 96, at 2333-37; *see also id.* at 2317-20 (arguing that programs with different code but identical behavior are market substitutes). *But cf.* Abramson, *supra* note 95, at 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 engineer 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 111. I see no reason to doubt the value of "black-box testing" for that task.

[115] Lee Interview, *supra* note 77 (transcript at 1); *see* Murphree Interview, *supra* note 87 (transcript at 1); Stephenson Interview, *supra* note 90 (transcript at 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 Interview with Tom Bishop, Chief Technology Officer, Vieo (Jan. 28, 2003, Austin, TX) [hereinafter Bishop Interview] [transcript on file with author] (transcript at 2) (arguing that reverse engineering is easy when the ideas in a product are "what the customer wants"; Like

enterprise software product is not writing the code, but understanding the problem that needs to be solved.[116] 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.

That is not to say that producing a program that adopts a competitor's product features is always trivially easy. For one thing, there plainly are differing views in the industry. Although their views seem to me against the weight of the evidence, some apparently knowledgeable commentators continue to insist that reverse engineering in fact is quite difficult.[117] For me, the most powerful evidence of the ease of reverse engineering is the steps that software companies take to render it more difficult. Thus, all of the companies with whom I discussed the subject took some steps to ensure that their competitors would not have access to their software code.[118] The most common and time-honored tactic rests on the distinction between a program's more comprehensible "source code" and its less comprehensible "object code."[119] Thus, many commercial developers traditionally have kept the more comprehensible source code secret,

---

[graphic user interfaces], once you know that's what people want, easy to do."); Costello Interview, *supra* note 107 (transcript at 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, *supra* note 107 (transcript at 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, *supra* note 82 (transcript at 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.").

[116] Weghorst Interview, *supra* note 81 (transcript at 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.").

[117] *See* Mark A. Haynes, *Black Holes of Innovation in the Software Arts*, 14 BERKELEY TECH. L.J. 567, 568-73 (1999) (arguing that copyright's restrictions on reverse engineering slow the pace of innovation in the software industry).

[118] *See*, e.*g.*, Koontz Interview, *supra* note 106 (transcript at 1) (discussing the concern of game developers to prevent people from seeing their "under-the-hood" work).

[119] Basically, source code is the version of a program written in a computer language readily accessible to ordinary computer programmers. Before the program is run on a computer, it normally is compiled into object code, which is not generally comprehensible to even the most skilled computer programmers. *See* MICHAEL D. SCOTT, INTERNET AND TECHNOLOGY LAW DESK REFERENCE 559-61, 729-34 (2003); Mathias Strasser, *A New Paradigm in Intellectual Property Law? The Case Against Open Sources*, 2001 STAN. TECH. L. REV. 4, at 5-7. This distinction, of great importance in earlier years, apparently has become less important with the rise of Java-based programming. *See* D'Eath Interview, *supra* note 83 (transcript at 4); Urdahl Interview

distributing their products to their customers only in the form of less comprehensible object code.[120] 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 from competitors more readily than the servers of their customers can.[121] 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.[122] Those customers, inevitably bound by license agreements not to redistribute the software to third parties, would not – my interview subjects repeatedly insisted[123] – assist competitors in reverse engineering efforts.[124] 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."[125] A few discussed more low-tech approaches, such as delivering the code in a module secured by

---

[120] *See* Abramson, *supra* note 95, at 77: Strasser:7. Several of my interview subjects rely on that practice. Abbott Interview, *supra* note 110 (transcript at 8); Rightmer Interview, *supra* note 82 (transcript at 2); Sikora Interview, *supra* note 83 (transcript at 1). The "open-source" movement is in large part defined by its complete rejection of that practice – a program cannot qualify as open-source if the developer does not make the source code publicly available. For the authoritative definition of "open-source", see *The Open Source Definition* (Version 1.9) (2003) *available at* http://opensource.org/docs/definition.php (last visited May 22, 2003). *See generally* MARTIN FINK, THE BUSINESS AND ECONOMICS OF LINUX AND OPEN SOURCE (2003) (general discussion of the open-source business model). In recent years, even commercial developers have begun to make their source code more widely available, at least to reliable customers and business partners. *See, e.g.*, Microsoft Corporation, *Shared Source Licensing Programs* (2003), *available at* http://www.microsoft.com/resources/sharedsource/Licensing/default.mspx (last visited May 22, 2003).

[121] *See* Harding Interview, *supra* note 75 (transcript at 2).

[122] *See* Bishop Interview, *supra* note 115 (transcript at 3); D'Eath Interview, *supra* note 83 (transcript at 4); Rightmer Interview, *supra* note 82 (transcript at 2).

[123] *See* D'Eath Interview, *supra* note 83 (transcript at 4); Rightmer Interview, *supra* note 82 (transcript at 2) ("Our clients (Circuit City, Home Depot, and the like) won't do that."); Weghorst Interview, *supra* note 81 (transcript at 2-3) (commenting that the Fortune 500 companies that are his customers are more concerned about compliance than he is "because they're the ones with the deep pockets and the heavy exposure").

[124] The Federal Circuit recently has upheld a provision in a license agreement preventing reverse engineering. *Bowers v. Baystate Technologies, Inc.*, 302 F.3d 1334 (2002). *But cf. Vault Corp. v. Quaid Software Ltd.*, 847 F.2d 255 (1998). For a cogent argument for a contrary result, see David A. Rice, *Public Goods, Private Contract and Public Policy: Federal Preemption of Software License Prohibitions Against Reverse Engineering*, 53 U. PITT. L. REV. 543 (1992) (arguing that federal intellectual-property policy preempts state contract law that would prevent reverse engineering); *see also* Pamela Samuelson & Suzanne Scotchmer, *The Law and Economics of Reverse Engineering*, 111 YALE L.J. 1575, 1660 (2002) (making a similar argument for economic reasons).

[125] D'Eath Interview, *supra* note 83 (transcript at 6).

a lock to prohibit access by the customer.[126] The import of those efforts, of course, is that copyright protection is not effective to prevent reverse engineering by competitors.

Indeed, a number of my interview subjects went even farther. Those executives argued that such efforts are wasteful, because access to the actual code is not useful for most types of reverse engineering. Those executives argued that the need to integrate the reverse-engineered software into the operating environment of the competitor would make it counterproductive to start from the code of the originating innovator.[127] It normally would be easier, they say, to start from scratch writing code to implement the observed functionality, than it would be to start from the existing code and alter that code to match the reverse engineer's existing environment. From that perspective, the emphasis on code that is at the foundation of copyright protection renders it entirely irrelevant to the protection of the startup firm's work in process.

B. The Role of Collateral in Startup Lending

There is one area in which investors in startup firms show particular interest in copyright protection. That area is investments by lenders (principally banks). As I have discussed in my earlier work, those lenders often take a security interest in copyrights held by the portfolio firm.[128] At first glance, that practice is in considerable tension with the assertions in the preceding section about the general irrelevance of copyright protection in the process of attracting funding to software-developing startup firms.

A closer examination of the lender's practices shows, however, that they are in fact consistent with my thesis. The key is the purpose of the lender's security interest. As my previous article explains, the purpose of those security interests cannot plausibly be to ensure that the loan is collected through a repossession and foreclosure of the

---

[126] *See* Bishop Interview, *supra* note 115 (transcript at 3) ("We call it 'ASP in a box." Access to the box is limited. Hard to lift up the cover. They couldn't open the box w/o us knowing. * * * They can see what goes in and out of the box, but they can't replicate the brain. Just like Jack Nicklaus.").

[127] *See also* D'Eath Interview, *supra* note 83 (transcript at 7) (discussing how hard it would be to understand his product just by looking at the code); Interview with Rod Favaron, President and CEO, and Phil Gilbert, Executive Vice President and CTO, Lombardi Software (Feb. 27, 2003, Austin, TX). For a similar academic perspective, see Samuelson & Scotchmer, *supra* note 124, at 1613-15 (arguing that object-code reverse-engineering generally is not useful).

[128] Mann, *Software Financing*, *supra* note 59, at 175-76; *see* Interview with Marc Cadieux, Division Manager (Peninsula Region), Silicon Valley Bank (Feb. 6, 2003, Santa Clara, CA) [hereinafter Cadieux Interview] [transcript on file with author] (transcript at 11); Interview with Timothy Klitch, First Vice President, Comerica Incorporated (Nov. 15, 2002, Austin, TX) [hereinafter Klitch Interview] [transcript on file with author] (transcript at 1); Interview with Philip Wright, Vice President, Comerica (Dec. 3, 2002, Austin, TX) [hereinafter Wright Interview] [transcript on file with author] (transcript at 1).

copyright-protected software.[129] One thoughtful lender interviewed for this project explained:

> [T]hat's been the biggest lesson we've learned over the last couple of years: that reliance on intellectual property as an ultimate source of repayment is very unreliable, very high beta. So we've had to focus much more on the venture relationships that we have and making sure that we work towards a soft landing – not letting them just run into the wall and then hope that when the pieces fall to the ground that we can pick up the intellectual property and sell it for at least our loan balance.[130]

Even outside the liquidation context, sophisticated lenders recognize that copyright protection cannot really contribute to the value of a software startup company. For example, one lender explained that "the enterprise value associated with intellectual property is an important factor in the underwriting process, but I would say that it falls into a spot second, third, fourth, or maybe even fifth in terms of relative priority."[131] Rather, the principal source of repayment is out of future fundings by the venture capitalists that typically invest simultaneously with the bank lender.[132] Speaking specifically of early-stage lending, one lender went so far as to say that "the collateral is the equity investors."[133] The basic premise is that the lender is safe in relying on reputation if it focuses on repeat dealings with management teams[134] and venture

---

[129] Mann, *Software Financing*, *supra* note 59, at 175-76. My interviews for this project are consistent with that view. *See* Cadieux Interview, *supra* note 128 (transcript at 5-6) (discussing deterioration in recoveries generated by sale of IP collateral from the early 1990's until the present); David A. Jones, Executive Vice President and Chief Credit Officer, Silicon Valley Bank (Feb. 6, 2003, Santa Clara, CA) [hereinafter D. Jones Interview] [transcript on file with author] (transcript at 6) (explaining that his IP collateral "might as well be real estate in a ghost town"); Klitch Interview, *supra* note 128 (transcript at 1-2).

[130] Interview with Andy Enroth, Senior Credit Officer, Silicon Valley Bank (Jan. 9, 2003, Austin, TX) [hereinafter Enroth Interview] [transcript on file with author] (transcript at 3).

[131] D. Jones Interview, *supra* note 129 (transcript at 1); *see* Cadieux Interview, *supra* note 128 (transcript at 2) ("I would say in terms of intellectual property protection it might surprise you to learn that falls fairly low on the list compared to things like which VCs are in the deal, what space the company is in, how many other companies are already in that space, how much they want to borrow, who is on the board, how do we think they are going to do from a customer-traction standpoint."); Enroth Interview, *supra* note 130 (transcript at 1) ("Copyright protection [we're] not as focused on. There, it's really hard to garner whether there's intrinsic value or not. That to us would not be determinative as to whether a company was valuable or not.").

[132] *See* D. Jones Interview, *supra* note 129 (transcript at 6) ("[I]nvariably our source of repayment is the next round [of venture financing.").

[133] Wright Interview, *supra* note 128 (transcript at 3). *See also* Wright Interview, *supra* (transcript at 1) ("It is a little risky, but we are not only relying on our collateral. We lend money to people that we're comfortable with for some reason, whether it's because we've dealt with them or the management team or we're very comfortable with their venture capital sponsors.").

[134] A typical anecdote:

capitalists[135] that have behaved well in the dissolution of previous unsuccessful portfolio companies. Moreover, the large number of failed portfolio firms in recent years has provided venture capitalists and management teams the opportunity to display their reliability.[136]

> I have a client now who had one of the high-flying Internet companies. They had millions of dollars. Product never really sold. Shut it down. The first thing they did during the wind down was they said ahead of time when they still had cash, they paid off the bank early on, got us out of the way. He has the world's best credit. He's got another company now and we are very open to doing whatever we can for him, bending all the rules. Here is a guy who took care of us we're going to try to pay him back.

Wright Interview, *supra* note 128 (transcript at 1-2).

[135] *See* Enroth Interview, *supra* note 130 (transcript at 3) ("For us, this is all about relationships, so even if the company's technology is failing, if the investors still are supportive and think that there is an opportunity to sell a piece of it off or do whatever we are likely to be very very flexible."); Wright Interview, *supra* note 128 (transcript at 3) ("At the first stage, the collateral is the equity investors. People like Rob Adams [of Austin Ventures] that have done a lot of those deals. We're very comfortable doing them with him. * * * Out of the last 75 deals we've done with Austin Ventures we've lost $200,000 dollars or so but loaned X million dollars. [The loss rate] is a lot less than [is typical for our portfolio]."). Indeed, several of the lenders indicated that the relationships descend to a personal level – that a bank's decision whether to make a loan might turn not on the identity of the venture-capital firm, but on the particular individual at the firm sponsoring the investment:

> [W]e know that this particular partner has behaved or reacted the way that we would hope as a bank, and there are other funds – and Austin Ventures actually has treated us very well – and there are other funds that maybe the fund as a whole has treated us real well, certain partners, but maybe there's been one partner that has reacted in a completely unpredictable manner that surprised us. And banks don't like being surprised. So we'll remember that. And so when the next deal comes in that's funded by him it may be tough for some of the line guys who are more sales-driven but we'll remember.

Enroth Interview, *supra* note 130 (transcript at 5); *see* D. Jones Interview, *supra* note 129 (transcript at 15) ("[John] Thornton [at Austin Ventures] is the guy that I might almost give him a $500,000 check and say, 'All I want you to do is tell me within 24 hours of you putting the name on it – giving us somebody who you gave it to so I can put in on my books.'").

[136] *See* Enroth Interview, *supra* note 130 (transcript at 5) ("We've been through the bad times. And we know that this particular partner has behaved or reacted the way that we would hope as a bank * * *); D. Jones Interview, *supra* note 129 (transcript at 2) (noting that a prior failed company "wouldn't be the black flag that it might be for most lenders, because in our space a high percentage of companies do not succeed"). Lenders also report themselves to be constrained by reputational concerns not to move too swiftly in deciding to "MAC" a borrower – to accelerate the borrower's obligation to repay the debt based on a material adverse change in the borrower's financial condition. *See* D. Jones Interview, *supra* note 129 (transcript at 14-16) (explaining that reports of his bank MAC'ing borrowers are urban legends, and that concerns about future relationships are a substantial constraint in making those determinations); Klitch Interview, *supra* note 128 (transcript at 2) (explaining that his bank is less prone to MAC'ing borrowers than competing financial institutions).

Thus, the purpose of the formal collateralization of the IP is to deal with one narrow scenario,[137] in which the loan is not repaid and the borrower becomes insolvent. In that event, the lender wants to ensure that it has control of all of the assets – whatever they might be. The point of that control is to ensure that in a bankruptcy proceeding the lender will have the leverage to control that proceeding as effectively as possible.[138] If a specified asset existed – a copyright interest in a software program, for example – in which the lender did not have a perfected security interest, then the lender would not have priority (and control) with respect to the entire estate of the bankrupt borrower.[139]

From a bigger perspective, this suggests that copyright protection – at least from the perspective of the lender – is affirmatively perverse for the startup company. The difficulty is that under the current system, copyright protection has two attributes that together make it problematic for the lender. First, it attaches automatically – an interest in the developer's software arises under the Copyright Act automatically each day as revisions to the software are fixed in a tangible medium of expression.[140] Second, no matter how carefully the lender acts, the lender cannot obtain a reliably perfected security interest in the software.

For example, if the lender adopts the strategy of filing in the federal Copyright Office, it can file only with respect to registered copyrights.[141] At least under current cases, that would give it a perfected security interest in the registered copyrights, which should be enforceable against both later lenders and the trustee in bankruptcy.[142] The problem, however, is that federal filings are not available for unregistered copyrights.[143]

---

[137] *See* Klitch Interview, *supra* note 128 (transcript at 2) (noting that the Austin office of his bank had not had to do a liquidation in the preceding year (2002)).

[138] *See* Wright Interview, *supra* note 128 (transcript at 2) ("We just don't want to find out that there is some value and we're not protected.").

[139] To be sure, the lender would have some control over that asset, because it would be a derivative work over which the lender would have some control. The difficulty, however, is that the derivative work will have been prepared "lawfully" by the copyright owner, so a copyright interest will arise in the owner; if registered the lender will not have acquired that interest through its state-law security interest. *See also* Douglas G. Baird & Robert K. Rasmussen, *Control Rights, Priority Rights, and the Conceptual Foundations of Corporate Reorganizations*, 87 VA. L. REV. 921 (2001) (arguing that the core issues in corporate reorganizations are about control of the assets rather than priority of distribution).

[140] *See* Copyright Act § 102(a). This is a major change from the Act of 1909, under which protection attached only when the work was "published." *See* ROBERT A. GORMAN & JANE C. GINSBURG, COPYRIGHT: CASES AND MATERIALS 384-89 (6th ed. 2002).

[141] Copyright Act § 205(c)(2).

[142] *See In re* World Auxiliary Power Co. (Aerocon Engineering, Inc. v. Silicon Valley Bank), 303 F.3d 1120, 1125-28 (9th Cir. 2002).

[143] *See In re* World Auxiliary Power Co. (Aerocon Engineering, Inc. v. Silicon Valley Bank), 303 F.3d 1120, 1128-31 (9th Cir. 2002).

In addition, as a practical matter, the lender cannot cause *all* of the copyrights to be registered. As mentioned above, copyright protection attaches each day to any revisions to the software that are sufficiently substantial to involve protectible expression – a minor threshold doubtless surpassed by any cognizable amount of work. The time and expense of registering the software each time it is changed – perhaps each day – is not something on which a lender reasonably could insist during the course of active software development.[144]

On the other hand, suppose the lender adopts the strategy of filing in the state UCC records. At least under current cases, that should give the lender a perfected security interest in unregistered copyrights.[145] The problem with that strategy is that it is difficult for the lender to ensure that *none* of the copyrights is registered. If the borrower later happens to register the software, the lender's security interest probably becomes ineffective at the time of registration. Moreover, unless the lender files again in the federal records at the time of registration – something it can do of course only if it is aware of the registration in advance – it risks losing priority to a later lender.[146]

That all would be fine if the copyright protection – like the patent protection – itself were a valuable asset to the startup. Then, the accretion of that protection to the borrower's pool of assets would be something valuable in which the lender understandably would want an interest. As discussed above, however, copyright protection really does little or nothing of value for the startup. Moreover, if the lender's only reason for wanting a perfected security interest were to make sure that it is perfected in the entire balance sheet, the lender's task would be simpler if the system were reformed in a sensible manner. For example, the system could work if copyright protection did not attach until the software was registered and if the security interest in the unregistered software automatically attached to the registered work that resulted from the unregistered work. More boldly, the system could be reformed so that all security interests in copyrights were taken under state law.[147]

C. The Role Of Copyright In Later-Stage Firms

The previous part's skepticism about the utility of copyright for startups should not be read to suggest that copyright is not important in the software industry. On the

---

[144] *See* Mann, *Software Financing*, *supra* note 59, at 146-47, 150-51.

[145] *See In re* World Auxiliary Power Co. (Aerocon Engineering, Inc. v. Silicon Valley Bank), 303 F.3d 1120, 1128-31 (9th Cir. 2002).

[146] *See In re* World Auxiliary Power Co. (Aerocon Engineering, Inc. v. Silicon Valley Bank), 303 F.3d 1120, 1131-32 (9th Cir. 2002) (frankly noting the problem).

[147] The legislative history of the Copyright Act suggests that the statutory requirement of federal filings was intentional (however impractical), so the removal of that requirement would be a significant policy retreat. *See* Harold R. Weinberg & William J. Woodward, Jr., *Legislative Process and Commercial Law: Lessons from the Copyright Act of 1976 and the Uniform Commercial Code*, 48 BUS. LAWY. 437, 453-66 (1993)

contrary, 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.[148]

1. 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 the startup's product: its customers. Thus, where patent and trade secret protection are much more important in limiting the ability of competitors – horizontal peers in the industry – to take technology from the innovator, copyright is much more important in limiting the ability of customers to obtain the product without paying the product's owner.[149]

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. As discussed above,[150] there are reliable methods of limiting piracy in that market, including license terms that are effective to prevent authorized distributees from transmitting the software code to third parties. Key to the effectiveness of those provisions 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.[151] Even in this country, it is apparent that consumers commonly violate the terms of license agreements, copying and

---

[148] I expect to explore this topic further in a paper analyzing the types of cases brought under the Copyright Act. For example, data from the Northern District of California for 2002 shows that out of the 90 copyright cases, 27 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. Of the remaining seven, four involved claims against competitors for wrongful reverse engineering, and three arose out of contract disputes between web-site developers and their customers.

[149] *See supra* note 109 and accompanying text (articulating a distinction between vertical and horizontal protection).

[150] *See supra* pages 22-24.

[151] *See* D'Eath Interview, *supra* note 83 (transcript at 7) ("That answer is going to be very different for somebody who is selling a retail product."); Kelley Interview, *supra* note 107 (transcript at 2) (emphasizing importance of copyright protection for mass-market software); *see also* Karjala, *supra* note 96, at 67 ("[O]nce * * * programs are distributed in object-code form, they can be copied almost without cost in large numbers.").

transferring software in ways that – if the licensor software publisher had its way – would require payment from the new user.

Executives generally recognize that in other countries the problem is a serious one even in the enterprise-software context.[152] 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 software distributed in the country is distributed through lawful channels.[153]

For several reasons, copyright is the only effective IP protection against piracy.[154] For example, even if the pirated software is 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

---

[152] *See* Adams Interview, *supra* note 89 (transcript at 1) (discussing piracy of Lotus in Italy); Interview with Tim Cranton, Senior Attorney, Microsoft (Feb. 4, 2003, Redmond, WA) [hereinafter Cranton Interview] [transcript on file with author] (transcript at 1); D'Eath Interview, *supra* note 83 (transcript at 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."); Telephone Interview with Annmarie Levins, Associate General Counsel, Worldwide Sales Group, Microsoft (Feb. 25, 2003) [hereinafter Levins Interview] [transcript on file with author] (transcript at 1) (discussing major businesses engaged in piracy in Eastern Europe); Sikora Interview, *supra* note 83 (transcript at 1) (offering examples of Russia and China); *see also* Cranton Interview, *supra* (transcript at 2) (suggesting that enforcement of copyright protection over the Internet is similar to enforcement in a developing country).

[153] *See* IDC & the Business Software Alliance, *Expanding Global Economies: The Benefits of Reducing Software Piracy* (Apr. 2, 2003) (copy on file with author) (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%).

[154] *See* Interview with Dan Crouse, Deputy General Counsel, Patents, Microsoft (Feb. 4, 2003, Redmond, WA) [transcript on file with author] (transcript at 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, *supra* note 96, at 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, *supra* note 124, at 1613 ("Copyright law protects programs from the cheapest and most rapid way to make a directly competing identical product, namely, copying program code exactly.").

the threshold of patentability.[155] 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.[156] 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 even a statute for criminal patent infringement.[157]

2. Pre-Market Protection

Copyright also plays a role even before a product goes to market, in helping a firm prevent technology from leaking out through the actions of employees and business partners. The most 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.[158] Yet, it is one thing for employers to allow the cross-pollination of employees and their human capital and experiences. It is quite another for their employees to take substantial chunks of "product" out the door with them and sell that product from their new company.

Copyright protection is central to the prevention of that activity. Again, patent protection is relatively ineffective, because of the litigation costs and uncertainty of such litigation. Still, in cases in which the employees attempt to reuse a substantial amount of

---

[155] *See* 35 U.S.C. §§ 101-103 (principal conditions for patentability); D.C. TOEDT, THE LAW AND BUSINESS OF COMPUTER SOFTWARE §§ 3.3 - 3.4 (2002) (discussion of patentability requirements as applied to computer software).

[156] *See* Copyright Act § 102(a) (extending copyright to "original works of authorship fixed in any tangible medium of expression"); ROGER E. SCHECHTER & JOHN R. THOMAS, INTELLECTUAL PROPERTY: THE LAW OF COPYRIGHTS, PATENTS AND TRADEMARKS § 3.1.2 (2003) (discussing "minimal creativity" requirement for copyright protection).

[157] For example, DOJ data shows 25 indictments of 73 defendants for criminal copyright infringement in 2002, but not a single indictment for criminal patent infringement. http://www.usdoj.gov/ag/annualreports/pr2002/AppendixC.htm. For similar views from a former prosecutor, see Levins Interview, *supra* note 152 (transcript at 1); *see also* Interview with Thomas C. Rubin, Associate General Counsel, Microsoft (Feb. 4, 2003, Redmond, WA) [hereinafter Rubin Interview] [transcript on file with author] (transcript at 2) (similar view from executive at Microsoft).

[158] *See* ANNA LEE SAXENIAN, REGIONAL ADVANTAGE: CULTURE AND COMPETITION IN SILICON VALLEY AND ROUTE 128 (1994); 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).

code from their previous firm,[159] copyright law provides a simple and effective remedy against the new firm.[160] The importance of that constraint in the system is evident from the discussions of corporate counsel about their diligence with respect to new employees[161] 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 contamination.[162] 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.[163] Although the participants in that process are free to use contracts to define the rights each of them has in the various portions of

---

[159] "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, *supra* note 157 (transcript at 2); *see* Beauchamp Interview, *supra* note 107 (transcript at 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").

[160] The injunctive and criminal remedies discussed above also are important in that context. *See* Rubin Interview, *supra* note 157 (transcript at 2); Interview with Derek Witte, General Counsel, Silicon Valley Bank (Feb. 6, 2003, Santa Clara, CA) [hereinafter Witte Interview] [transcript on file with author] (transcript at 5).

[161] *See* Witte Interview, *supra* note 160 (transcript at 5):

> [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!

[162] 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, *supra* note 77 (transcript at 1).

[163] *See* Rubin Interview, *supra* note 157 (transcript at 1).

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.[164] 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."[165]

V. The Role of Patents

This part is the core of the paper. It relies on evidence from the interviews to discuss as precisely as possible how patents contribute to the ability of a software startup to obtain financing. It falls into three sections: the limited value of patents for startups, the increasing value for firms as they move beyond the startup stage, and the policy implications of those findings.

A. The Limited Value of Patents for Startups

The first point is the strongest one: what patents do not do. Specifically, patents are not often useful in helping the early-stage software company demonstrate the sustainable differentiation from its competition that is the focus of the venture investor.[166] There are several explanations for that point. Some of the problems arise from general difficulties in obtaining software patents that actually have value, difficulties that afflict firms of all sizes. I will say more about those problems below when I discuss the role of patents for growing firms, but it seems important to introduce those problems here. Generally, the interview subjects argued that in most cases a competitor could implement most of the aspects of a software product that a patent might protect without infringing the patent.[167] That assertion illustrates the poor match between patents and products in the industry: it is difficult to patent an entire product in the software industry, because

---

[164] *See* Rubin Interview, *supra* note 157 (transcript at 1-2) (discussing Copyright Act §§ 502, 504).

[165] Rubin Interview, *supra* note 157 (transcript at 2).

[166] I provide some quantitative support for that assertion in the companion paper with Tom Sager. The data analyzed in that paper indicates – with the limited exception of startup software firms that have been spun off from larger firms along with a patent portfolio – that very few software startups have patents when they receive early funding, and also that only a small portion of the firms that ultimately receive patents had even applied for those patents before they received their initial funding.

[167] *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.").

any particular product is likely to include dozens if not hundreds of separate technological ideas. Thus, it may take a large number of novel ideas – and patents – to build a defensible barrier around a product in the software industry. A biotech startup, by contrast, more often might build a defensible barrier around its product with one patent or only a few patents on the relevant composition or process. Another problem is that software 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.[168] Litigation at that point will involve efforts by the patentholder to use the patent to challenge technology of a subsequent generation where the application of the patent may be less clear.[169]

Those problems, however, are not necessarily any more of an obstacle to small firms than they are to large firms. There are some problems, however, that make it particularly difficult for early-stage companies to employ patents effectively. For example, even if an early-stage company had a patent, it is most unlikely that it would be an appropriate use of its resources to commence litigation to enforce a patent against a competitor.[170] That is particularly true when the competitor is a large firm:

> Most business people I know, particularly in young companies, the last thing they want to get involved in is a lawsuit. You run into that a lot where big companies will come in and work around your patent and just dare you to sue them. "We will rain lawyers on your head and tie you up in court for the next ten years." * * * *
>
> In the venture world, very seldom do disputes get down to the lawsuit stage, because nobody wants to spend the time and these young companies can't afford to go through all that. You think about it, most of the lawsuits seem to happen among the big companies who have teams of lawyers and can afford to go at each other.[171]

---

[168] *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).

[169] *See* Cohen & Lemley (2001).

[170] *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.").

[171] Murphree Interview:1; *see* Abbott Interview:5 (discussing resource constraints on early-stage patent enforcement); 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

A somewhat different concern about suing a large firm – related to the discussion of patent thickets below – 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. As one developer said: "IBM probably could sue us on 20 patents if they looked hard at what we do. But we don't want to have that relationship with them. Legal fees aside, we could lose everything."[172]

An important aspect of that point is the sense that there is something about firm culture that is degraded when a firm sinks to the level of relying on licensing revenues instead of developing its own product. As one experienced executive put it:

> I think a lot of companies made a mistake and lost their way and did that – Texas Instruments, Xerox, National Semiconductor. Those companies are making more money from licensing than from their business. Engineers don't go to work to do that. They want to make a better product or service. As soon as Xerox, Nat[ional] Semi[conductor], and TI and those guys started being really aggressive about patent licensing, that was the beginning of the end.[173]

Interestingly enough, that sentiment was expressed even at the two software firms that I interviewed that rely heavily on licensing revenues (neither of which, as it happens, are startups). Those firms emphasized efforts to maintain a product-centered culture emphasizing production of the firm's own products.[174] 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.[175]

---

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.

[172] 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").

[173] Van Arsdale Interview:2. In response to a question, he emphasized that IBM is *not* a counter-example to that reasoning: "If they are in a market they want to compete like the rest of us. They just happen to have a huge asset to go and enforce patents – as do we – and it gives you a huge asset to go and trade with other companies that are of a like mind." *Id.*

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

[175] Thomas Interview:3.

Indeed, at the one firm that had a major licensing program, the entire licensing program was entrusted to off-site third-party professionals, so that it would not interfere with the focus of the on-site software engineers.[176]

Thus, as suggested above, many of my interview subjects insisted that the venture investor is likely to focus on other concerns related to the company and the sector in which it operates or proposes to operate. Indeed, many investors (and developers) emphasized that a focus on patents in fact can be damaging to the 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.[177] 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."[178] 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."[179] 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."[180]

This presents the young company 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 future possibility that the firm *might* at some point grow to the point where the IP is useful.[181] 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

---

[176] Thomas Interview:4.

[177] *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.").

[178] Subhedar Interview:4.

[179] 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).

[180] Bishop Interview:2.

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

seeds of its inevitable failure if it survives to a later stage.[182] On the other hand, it must not spend so many resources that the company fails before it reaches the point where that investment can be recouped.

Firms have developed a number of strategies to deal with that difficult problem. Some involve using half-measures to protect the IP, such as filing provisional applications,[183] or omitting standard practices related to documentation of the work of engineers.[184] Those kinds of 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 grown to the point where they have substantial revenues it often does not seem appropriate to devote the resources necessary to ensure that all of the firm's innovations are patented.[185] Others – it must be said that executives with prior experience at large IP-sensitive firms like IBM[186] 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.[187] If anything is clear, it is that the difficulty of this strategic choice, coupled with the difficulty in accurately predicting the future prospects for their products and their IP, doubtless leads to many of the failures in the industry.[188]

---

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

[183] *See* Beauchamp Interview:3 4 (discussing use of provisional applications); Weghorst Interview:4 (same).

[184] *See* Weghorst Interview:6 (discussing deferral of documentation of inventions to limit up-front costs of IP protection). 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.

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

[186] 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.

[187] *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).

[188] *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.").

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 (or might not) provide.[189] In short, standard industry practice now views examination of that market power as central to careful due-diligence practices.[190] 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.

B. The Increasing Value of Patents for Growing Firms

As the discussion above suggests, it seems plain that patents are valuable to software firms at some stage in their development, even if it is not at the earliest stage, when the firm is still struggling to develop its first product. This section examines that question, illustrating how patents are valuable to software firms while they are still quite small, as they proceed through the stages of the venture-capital process.[191] To structure the discussion, I start with the quantitative evidence presented in detail in the companion paper.[192] That evidence shows two things of relevance to this discussion. First, even controlling for industry group and location, the number of rounds of financing that a firm receives is related at the 5% level to whether the firm has a patent. Similarly, the final status of the firm (acquired, defunct, existing, public) is related to the number and existence of patents at the 1% level. Those findings suggest that patents matter in some way. At the same time, the explanatory power of those findings is quite small – the adjusted R-squared values range from 0.5% to 1.3% in the various formulations of the test. That suggests that whatever the value is, it is not the leading indicator of the firm's

---

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

[190] *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).

[191] *See* Gauer Interview:1-2 ("I've never invested in a software company that had a patent when I invested in it. I have had companies that were working on patents that four years later became patents."); Lee Interview:1 (indicating no concern about the existence of patents at the "seed" or "A" stage, but considerable interest in them for firms farther along). Again, this finding is consistent with the results of the Cohen et al. cross-industry surveys, which reported a significant correlation between firm size and the effectiveness of patents in appropriating the value of innovation. Cohen et al. (2000):12.

[192] Mann & Sager (2004).

success, but at best an episodic or indirect factor. That builds on the discussion in the previous section, indicating the consistent perspective that IP is of at best secondary importance.[193]

1. 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. As discussed above, it is common in the software industry to report that software patents are used defensively, rather than offensively.[194] In conversation, this assertion tends to be coupled with the problem that software patents tend to be relatively weak devices for appropriating the value of a software-related invention.[195] Although there is some truth in both of those points, they are gross 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 interview sample three small companies – Applied Science Fiction,[196] Bluecurrent,[197] and Forgent[198] – have obtained substantial revenues in that way. I have not located industry-wide statistics that 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. Those transactions seem to me particularly noteworthy given the difficulties those firms face in enforcing patents against large firms with their own patent portfolios. As discussed above,[199] 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.

---

[193] *See* Kielb Interview:1 (even in areas in which patents are useful, "it is the ability to build something that is usable and solves a problem that produces 80% of the value").

[194] *See supra* notes 170-176 and accompanying text.

[195] *See supra* notes 167-168 and accompanying text.

[196] 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, 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.

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

[198] 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.

[199] *See supra* note 172.

More generally, it seems clear that the received wisdom that patents are not useful to appropriate software-related inventions is quite overstated. Two separate points are important. The first point 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 good reasons why a firm might want to wait until quite late in its development before it advertises 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 general perception that patents tilt the playing field in favor of large incumbent firms to the disadvantage of small firms,[200] patents in this context afford a unique opportunity to the small startup. 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.[201] For the smaller firm, however, the ability to prevent incumbents like 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."[202]

On that point, it is instructive to think of the most salient uses of software patents. The most famous successful uses of software patents are plainly not use by giants to stamp out incipient competition. On the contrary, the most famous incidents are 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 relatively tiny firms, and a similar attempt by InterTrust to assert rights to digital rights management technology important to a variety of Microsoft products.[203] The general point is consistent with recent empirical work suggesting that patents held by small firms are more likely to be litigated than patents held by large firms.[204]

---

[200] I discuss that perception in detail in subpart VI(C).

[201] 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.

[202] Thomas Interview:2. For a general analysis of that point, see Barnett (2003).

[203] 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.

[204] Allison, Lemley, Moore & Trunkey (2003).

The second point is that the ability of a patent to appropriate the value of an invention will vary along a number of dimensions. One of the most commonly mentioned dimensions in my interviews relates to the nature of the particular innovation. Thus, it generally is regarded as bordering on futile to rely on a patent in which the innovation lies in a method of writing software code.[205] 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.[206] Interestingly, that distinction seems to cut in the opposite direction of the conventional wisdom that "process" patents tend to be less valuable than "product" patents.[207] In the software industry, a patent on the particular product tends to 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.[208]

Loosely related to that continuum is a variation from sector to sector. 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 whole sample was about 0.7 patents per firm. Several sectors, however, had markedly higher rates: 4.9 per firm in graphics and digital imaging,[209] 2.7 per firm in the small sectors of expert systems and natural language, 1.6 per firm in multimedia, and 1.0 per firm in the security sector. At the same time, some relatively important sectors had unusually low rates of patenting: no

---

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

[206] *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 the biotech executive that I interviewed 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.

[207] *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).

[208] Thomas Interview:2; Urdahl, Hill, Witek Interviews.

[209] That includes the 62 patents held by Flashpoint, the highest patenter in our dataset. Even excluding Flashpoint, there were about 1.35 patents per firm in that sector.

patents at all for the 25 firms in the email and internet software sectors, .03 in applications software, and .12 in software for financial institutions.[210] That variation is quite startling, because it is quantitatively discernible even though patents are thought to be less valuable for software generally than they are for hardware,[211] and even though patents are much less common in the software industry as a whole than in some other industries (biotech being the most commonly noted example).[212] 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 and lower rates of patenting associated with types of innovation less susceptible of appropriation by patent.

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 that the previous parts summarize in sustaining differentiation from competitors, the likelihood that patents can provide shelter for some firms is an important one. The shelter may be difficult to predict – because it must depend for the most part on the breadth of market protection a patent turns out to have by the time it is issued, something that is quite difficult to predict when the patent application is filed.[213] 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 is difficult to believe that this is not a part of what makes patents and their breadth an item of interest to investors.

2. Indirect Effects: Facilitating Investment

The previous section argues that the most important structural significance of patents with respect to small firms is that they have the potential to protect a successful product from appropriation by an incumbent competitor. It is clear from the interviews,

---

[210] 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.

[211] For similar sentiments, see 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).

[212] The companion paper with Tom Sager shows that only about 20% of venture-backed software firms have patents, while about 50% of venture-backed biotech firms. It also shows statistically significant variations in the likelihood of patenting among the different sectors of the software industry.

[213] Dinkin Interview

however, that patents have a number of more subtle and less direct effects on the flow of investment and acquisition dollars into the growing software firm. This section organizes those indirect effects under the four separate headings that follow.

(a) Barter for Cross-Licensing

The most prominent explanation is that patents will be useful as "barter" in cross-licensing agreements that the firm would enter into if it reached a sufficiently mature stage to be a significant player in the industry. The interview subjects 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 rising 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.[214] The likelihood that the firm will be called upon 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.[215] Nevertheless, a patent to offset IBM's potential claim is of little value until the day when IBM demands royalties. IBM typically does not ask for royalties until the firm is earning sufficient revenues to justify the inquiry.[216]

---

[214] *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.").

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

[216] *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").

(b) Signaling

The second benefit of patenting is a "signaling" effect.[217] Some say that patents provide a "signal" of engineering discipline and market understanding. The premise of that argument 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.[218]

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.[219] Notice, of course, that this use of patents says absolutely nothing about the actual uniqueness of the 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 in Part IV, many investors think that inadequate market analysis

---

For a good example from the hardware industry, one of my interview subjects discusses the early history of Dell, upon which 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.

I discuss in subpart VI(C) reasons why it is unlikely to be fatal to the firm if it does not have patents to cross-license. It suffices for the present discussion to illustrate why a patent would be a valuable thing to have in potential cross-licensing discussions.

[217] *See generally* Long (2002) (discussing circumstances in which patents can provide valuable signals).

[218] Beauchamp Interview:4.

[219] 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, 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'?").

and execution are among the most common reasons for the failure of startup companies.[220] A signal that suggests something positive on that score is a useful one.

(c) Facilitating the Codification of Knowledge

Although it was not raised in any of my interviews, I discern considerable merit in a closely related role of the patent that is apparent from Ashish Arora's recent writings on innovation. He argues that one of the key problems in transferring knowledge from one firm to another is the ability to convert tacit knowledge – which is difficult to verify or transfer – to codified knowledge, which readily can be evaluated and transferred to third parties.[221] Because a patent by definition – at least if it satisfies the statutory criteria – includes the knowledge necessary to allow a person knowledgeable 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.

(d) Facilitating Acquisitions

The last commonly proffered benefit of patents relates to acquisitions. As discussed below, 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.[222] 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.[223] Even there, however, it is clear that the sophisticated acquirer

---

[220] 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.

[221] ARORA ET AL. (2001):95-99.

[222] Claims of infringement might be more of a concern for a large firm because their revenue base is so much larger that even a low-percentage royalty claim could cause significant damages. The recent Eolas case against Microsoft is a textbook 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 rate 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.

[223] *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

will focus on the business proposition that the company offers to its customers and whether that business proposition makes sense apart from the IP that might protect the ability to deliver that proposition.[224] 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.[225]

That same explanation can be spun 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 take on the role of patents in acquisition characterizes them as valuable for "marketing,"[226] convincing the investors in public markets that the company's technology is valuable.[227] The idea here 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, 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.[228] Thus, the argument goes, they tend to rely (less thoughtfully) on the mere existence of patents in the company's portfolio.[229] That argument is made particularly with respect to protecting the downside

---

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).

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

[225] Van Arsdale Interview:3.

[226] 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.

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

[228] One developer offered the following description of his firm's patent:

> It was useful, not in the traditional sense. I don't think it could ever be sold by itself for something. But it allowed us to position ourselves as having a unique capability. We designed this for you. We created this. We patented it. We're the only ones. Was it immensely valuable in the business? I think it was more valuable from a marketing standpoint than anything.

Costello Interview:1; *see* Abbott Interview:5 (similar perspective).

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

in the event of a company's failure.[230] 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.[231] The truth of that view in any particular context is of course difficult to assess.

3. Summary

In sum, the interviews suggest a highly particularistic and localized set of explanations for the value of patents. Because of the anecdotal and fragmentary nature of the information, the discussion is necessarily tentative. Some obvious avenues for further research suggest themselves. For example, a much better understanding of the claims about the significance of litigation and licensing could be obtained by analyzing data about those events. For example, it would be interesting to know something about the frequency with which software patents are litigated, the extent to which the litigation is offensive or defensive, and the relative size of the parties to the dispute.[232] Similarly, more detailed evidence about licensing practices would be most instructive about the value of patents as barter in licensing: we do not know such basic facts as how often firms voluntarily request licensing arrangements; how often licensing arrangements are a mechanism for settling claims of infringement; or how much firms pay for the licenses they need to operate their businesses.[233]

Still, even the limited information available here provides a framework for understanding the role of patents in small software firms. At the earliest stage, when a firm is trying to define and develop its initial product, the patents may have no current value. But that will change quite early in the life of the firm, as the patents begin to have a variety of useful attributes around the time the firm begins to market its products. The most important claim is that the patents can play an important role in preventing existing incumbents from appropriating the small firm's products wholesale. In some cases, probably not many, the patents may be so valuable as to produce direct revenues at an early stage from licensing to larger existing competitors. Finally, less significant as a

---

[230] *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.").

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

[232] I am in the early stages of collecting data for such a project. Allison, Lemley & Moore (2003) provides the best evidence to date on that question. That project examines the characteristics of a large class of litigated patents, which indirectly sheds light on the types of parties that bring litigation. I hope to provide a more focused and party-based examination of litigation in the software industry, designed to understand the dynamics of the offensive use of patents by small and large software firms.

[233] I see no obvious way to collect reliable data on those issues.

structural matter, but nevertheless pervasive, are a variety of indirect effects that help investors to direct their investments to the firms most likely to succeed.[234]

C. Policy Implications

The analysis presented above is incomplete in several obvious ways. Most obviously, it focuses only on the small firms in the industry. Although the industry is young, there are of course a large number of firms that are significantly larger than the firms on whom this study focuses. Still, given the importance of that part of the industry to innovation, the analysis does support some tentative understandings about the much-debated policy questions related to the growth of patents in the industry.

The most important question of course is whether patents are good for the industry. In its broadest form, that is an almost unanswerable question. For example, it is not easy to determine empirically or analytically whether the pace of innovation in any particular industry at any particular time is optimal, hyperoptimal, or suboptimal. Still, most scholars have accepted as a working premise that software patents would be good if they promoted an increase in innovation and bad if they led to a decline in innovation. Moreover, even the narrower question – whether patents lead to more or less innovation – seems empirically intractable, given the difficulty in determining the relative weights of the positive effects (increased incentives to development) and negative effects (the potential for decreased access to patented technology). The question would be even more complicated if we took into account broader imperfections in the economy that cause the incentives of the individual firms to diverge from optimality.[235]

Having said that, this paper does provide considerable information that bears on those questions directly. First, the discussion in the previous parts makes it possible to situate the software industry as it currently exists within the multi-faceted theoretical discussion about differing types of innovation. That discussion suggests a key institutional question that drives the net effectiveness of patenting in the software industry: have transactional patterns developed that limit the costs of patents by making cross-licenses of important technology readily available at reasonable costs? If they have not, then patents probably are a drag on innovation in the industry. Second, even if those arrangements exist, are patents sufficiently valuable to those that hold them that they attract investment in the industry? If patents are not sufficiently powerful to alter the sectors into which investments in innovation are made, then it is difficult to believe that the transactions costs of the system are justified by the benefits.[236] Looking at those questions, the evidence suggests – albeit tentatively – the likelihood that patents are a

---

[234] The idea that patents might be especially useful to small firms ties in well with the persistent policy concern that our patent system should be designed in a way that would protect small inventors. Lemley & Chien (2003):1299.

[235] Markovits (1975).

[236] For a powerful demonstration that patents can alter the direction of innovation, see Moser (2003) (relying on an extensive dataset from 19th century international exhibitions).

positive force for innovation in the industry. Together, as I explain in more detail below, those points strongly undercut the facile claims of a stifling patent thicket that have received so much attention in the press. On the contrary, if anything, the principal benefit of patents seems to be to shelter small firms against the potential domination of incumbent firms.[237]

1. Patents and Theories of Innovation

For the last decade, the dominant strain in the rich literature on the effect of patents has been that patents work differently in different kinds of industries. That literature has tended to analyze the question in several different ways. Some scholars (most prominently Merges and Nelson in papers written by themselves, with each other, or with others) have focused on the question of innovation – analyzing different ways in which innovation proceeds in different kinds of industries, and then inquiring whether patents seem to foster or retard different kinds of innovation. Another tack has started from the other side of the question, starting from a theory of the effects of patents, and then inquiring whether those effects would tend to maximize innovation. With respect to the software industry as it currently exists, both of those approaches suggest the same answer: that any possible positive effects of patents are dependent on arrangements that make inventions widely available throughout the industry.

Starting with the first method of analysis, Merges and Nelson identify four different "industrial patterns of technical advance": discrete invention, cumulative technologies, chemical technologies, and science-based technologies.[238] Within that framework, it seems plain that software is a "cumulative" technology,[239] in which "today's advances build on and interact with many other features of existing technology."[240] They offer two major conclusions about innovation in those industries: that such industries develop most rapidly when there are multiple and competitive sources of invention rather than a single source, and that the adverse effects of strong patents or patent portfolios can be mitigated by cross-licensing schemes.[241]

The second method of analysis considers the purpose of patents. The seminal paper in that literature is Plant's 1934 paper, which sets out the basic "reward" or "invention motivation" theory of patents – that patents might but might not, induce innovation by providing monopoly profits to the holder of the patent.[242] It is clear that

---

[237] Jonathan Barnett argues that this generally is true, largely because of the relatively limited ability of small firms to take advantage of other techniques for appropriating the value of their inventions. Barnett (2003). It is not clear to me, however, that the literature presently includes a plausible example of such an outcome in an existing industry.

[238] *See* Merges & Nelson (1990):880; Merges & Nelson (1994):6-8.

[239] *E.g.*, Burk & Lemley (2003):82-90; Cohen & Lemley:40-42.

[240] *See* Merges & Nelson (1990):881; FTC Report:44-45.

[241] *See* Merges & Nelson (1990):908.

[242] *See* Plant (1934); Mazzoleni & Nelson (1998):1033-38.

this is an important benefit of patents in some industries,[243] although the software industry in its current form probably is not one of them.[244] The other major benefit of longstanding notice in the literature – said to be inherent in the bargain reflected in the patent statute[245] – is the idea that patents enhance innovation by disseminating information about inventions.[246] The idea is that the disclosure required by the patent statute[247] enhances the spillover effects of inventions by increasing the likelihood that people throughout the industry will learn about the invention rapidly. Whatever the merits of that possibility generally, it is unlikely to have significant relevance to the software industry. As others have noted, the Federal Circuit in software cases has interpreted the disclosure requirement in such a way as to minimize the likelihood that disclosures will require information that is directly useful to competing inventors.[248] Moreover, given the rapid pace of innovation, it will not often be the case that

---

[243] *See* Cohen et al. (2000):24.

[244] There is of course the phenomenon of patent holding companies – which have no operating products and exist solely to collect licensing revenues for patents that they (typically) have purchased from inventors. *See* RIVETTE & KLINE (2000):135 et seq.; Dinkin Interview:11; Gauer Interview:3 (noting that problem); Sikora Interview:1 (same); Subhedar Interview:2 (same); Witte Interview:1 (same); *see also* Burk & Lemley (2003) (characterizing such companies as "trolls"). Although this might change, my perception is that the phenomenon at the present is sufficiently small that it does not affect the structure of the industry as a whole. The interview references discussed above leave me with the view that those companies are more in the nature of a noisome gadfly than any serious threat to innovation. Even my interviews at Microsoft (the target of most of that litigation) did not reflect any serious concern about that phenomenon.

Indeed, those firms seem to me to have the potential to play an important role for patents for which commercialization requires extensive capital investments beyond the liquidity of the patentee. Whatever the merits of the dispute, the Eolas litigation with Microsoft illustrates how such a firm might be the only way for an inventor (the University of California in that case) to obtain the capital that was necessary to ensure that the inventor could obtain a return on a patent that has long-term value. Thus, in a case where the "efficient" outcome would be for the patent to be acquired by a major firm in the industry, the possibility that the patent might be transferred to a "troll" – with extensive litigation resources – helps to ensure that the market for acquisition of the patent (or the firm that holds it) pays "fair" value for the patent.

[245] In the Court's words, "the patent system represents a carefully crafted bargain that encourages both the creation and the public disclosure of new and useful advances in technology, in return for an exclusive monopoly for a limited period of time." Pfaff v. Wells Electronics, Inc., 525 U.S. 55, 63 (1998).

[246] *See* Mazzoleni & Nelson (1998):1038-40.

[247] Patent Act § 112.

[248] Burk & Lemley (2003); FTC REPORT:49. As Dennis Karjala has pointed out to me, the concerns about those cases limiting disclosure can be overstated. The decisions do not, to be sure, require disclosure of the source code. But if the patent is on the functionality of the software, and if the source code is thought to be easily replaceable (as I argue it is), those decisions might not be an important obstacle to disclosure.

information disclosed in a patent application filed years earlier will be of immediate value to competitors in the industry.[249]

Turning from those venerable frameworks, the first important modern theoretical analysis of the benefits of patents is the "prospect" theory articulated by Kitch, which argues generally that a prospect would be developed most efficiently if one person controls it.[250] Suzanne Scotchmer in particular has extended the sophistication of Kitch's analysis with respect to industries of cumulative innovation. Her work (among others) has contributed to the development of what Mazzoleni & Nelson generalize as an "orderly development" theory of patents.[251] In a series of papers (by herself and with various co-authors), Scotchmer's work recognizes that inventors in such an industry must have some control over later innovation or they will have an inadequate incentive to innovate themselves.[252]

The key problem is how to provide that control without unduly limiting access to later innovators. Kitch's analysis rests on the implicit premise that this problem is particularly important in areas characterized by broad or pioneer patents. Later scholars that have developed the theory more broadly agree with that point – because its principle focus is balancing the benefits of the reward theory against undue harm to later innovators.[253] Thus, there is some reason to doubt that the theory is directly applicable to the software industry, where the kinds of broad pioneer patents that those authors discuss

---

[249] *But see* Cohen et al. (2000):29; *see also* Gruner (2000):1007-13 (arguing that software patents increase rate of disclosure of innovation and lower costs of patent races).

[250] Kitch (1977). For a thoughtful updating of Kitch's prospect theory, see Kieff (2001); *see also* Gruner (2000):1023-28 (discussing the benefits of centralized control of exploitation of an invention).

[251] The term is from Mazzoleni & Nelson (1998):1042-44.

[252] Scotchmer (1991):31; Green & Scotchmer (1995); Samuelson & Scotchmer (2002). The first two of those papers argue that patents should last longer when innovation is spread across many firms (as it commonly is in the software industry). Of course, the rapid pace of obsolescence in the software industry suggests that the patent term already exceeds almost all of the period in which most patentable inventions will have a useful life. *See* Burk & Lemley (2003):106 n.182 (discussing proposals to shorten patent term for software).

A recent paper by Bronwyn Hall builds on that literature to argue that the standard of nonobviousness should make it harder to obtain patents in industries characterized by rapid and cumulative innovation. Hall (2003). That is consistent with the suggestions in Cohen & Lemley (2001) (discussed *supra* note 12 and accompanying text).

[253] *See* Mazzoleni & Nelson (1998):1042-44

are harder to find.[254] In any event, all agree that the key question is whether the industry can respond to such patents by developing low-transaction-cost patterns of licensing.[255]

2. TheMyth of the Patent Thicket

If the theories suggest that the key factual question is the ability of industry players to obtain access to technology, they do nothing to allay the concerns of the critics of the software patent. Indeed, the putative limitations on access to technology are the ground on which the most prominent criticisms of software patents have rested. Larry Lessig presents it forcefully as a matter of truth that the proliferation of software patents has created a patent "thicket" (Shapiro's term)[256] or an "anticommons" (a term Lessig draws from Michael Heller's work with Becky Eisenberg[257]). This concern also pervades James Bessen's work (by himself and with other co-authors).[258] Specifically, the idea is that there are so many overlapping patents in the industry that potential innovators cannot readily obtain the approvals necessary to conduct their research.[259] The analogy is to a Moscow apartment that sits vacant because of the inability of any particular user to obtain consents from all of the various parties with interests in the apartment.[260]

---

[254] *See* Burk & Lemley (2003):153-156 (suggesting pioneer-patent problem is not important in software industry because its "inventions are characterized by more incremental improvements").

[255] *See, e.g.*, Cohen et al. (2000):25-29; Lemley (1997):1061-67; Merges & Nelson (1990):908; Scotchmer (1991):33-34. For an important recent contribution from a more general perspective, see Wagner (2003) (arguing that increases in appropriability generally are likely to increase the amount of information in the public domain, not decrease it).

[256] I use the term from the extended discussion in Shapiro (2001). That discussion is considerably more guarded. He simply notes the possibility, a perspective with which I agree entirely.

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

[258] *See* Bessen (2003); Bessen & Hunt (2003); Bessen & Maskin (2002). None of Bessen's relevant work is yet published. I respond to it in detail here because it is the principal empirical authority that Lessig adduces to support his arguments. 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. For a thorough criticism of the Bessen & Hunt paper, see Hahn & Wallsten (2003).

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

[260] Heller (1997).

The idea of a "thicket" or "anticommons" in the software industry is difficult to credit.[261] When raised in my interviews, that thesis universally was rejected.[262] The basic premise of the model is that assets will go unused because of the costs of obtaining the permissions necessary to use them.[263] As discussed above, there is 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 power to say anything usefully 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.[264] 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 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 – that there would be *something* in IBM's portfolio that their product infringed.

But that posed no significant concern for them. It is perhaps an artifact of the particular history of the industry, but IBM has firmly set a course of *relatively* lenient

---

[261] My reactions are primarily based on my understanding of the history and practice of the software industry. As a more general matter, consistent with my understanding of that industry, 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).

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

[263] *See, e.g.*, Buchanan & Yoon (2000):4.

[264] 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.

enforcement of its IP rights. The lenience of its practices is generally 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 bring the eye of federal antitrust regulators back to its practices.[265] 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.[266] To the extent the anti-trust 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 money than a finding of infringement by a small party with a limited customer base that might be involved in litigation with IBM.[267]

That is not to say that IBM allows people to use its IP freely.[268] It is to say, however, that licenses to use its IP are freely available to all legitimate users.[269] 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 parties' own IP.[270] Thus, IBM has followed a consistent two-pronged strategy: attempting to gain as free access as it can to all IP in the industry

---

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

[266] CAMPBELL-KELLY (2003):109-10.

[267] *See* Dinkin Interview:8.

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

[269] 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 ET AL. (2001):235-36 (discussing requirements imposed on IBM by consent decrees in antitrust litigation with the United States); CHANDLER (2001):250 (same).

[270] 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 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).

(giving IBM the freedom to market and sell as freely as possible),[271] and at the same time generating a steady stream of revenues from its now massive software-patent portfolio.[272] As Rosemarie Ziedonis has shown, this strategy of heavy patenting is not at all uncommon for large firms like IBM in an environment characterized by fractionation of technology.[273]

That focus on freedom of action is a rational strategy for a large firm like IBM. Given 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 in fact free 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 nice 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.[274] Its recent adoption of an open licensing policy that resembles IBM's policy[275] 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 quite similar.[276]

The noted[277] 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.[278] As a theoretical matter, the paper is fatally flawed

---

[271] *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.").

[272] 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).

[273] Ziedonis 2004.

[274] I am not aware of *any* patent infringement litigation instituted by Microsoft. I am in the early stages of collection of data for an empirical project examining patent litigation by and against large software firms, which should shed more light on that question.

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

[276] I had two such interviews with representatives of Fortune 500 technology firms. Unfortunately, both requested anonymity with respect to that discussion.

[277] The paper is the principal empirical support that Lessig offers for his discussion of the subject.

[278] Bessen & Maskin (2002).

by its failure even to notice the literature discussed above, which indicates that the effectiveness of licensing will depend to a great degree on context.[279] To the extent that literature supports 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.[280] 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.[281]

The paper's empirical premises are similarly weak. Bessen and Maskin rest their model on the explicit assumption that firms will not be able to reach value-increasing licensing agreements to make technology widely available in the industry.[282] That is not only an assumption that sets them outside the general pattern of theoretical analysis; it also is one that makes their model implausible as an empirical matter. To be sure, they do present empirical evidence to support their model, but the evidence terminates with 1995 – a very early stage of the software industry – and they examine only firms that they identify as leading software patentees, rather than leading firms in the industry. Thus, their dataset includes companies like Ford, but excludes companies like Microsoft and Oracle. Thus, the dataset appears to include a large number of patents that may have little relevance to the competitive situation in the software industry and at the same time exclude many of the patents that have the plainest competitive significance. A plausible analysis of that question at a minimum must examine the effect of patents in major software firms.[283]

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."[284] 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

---

[279] *See* Merges & Nelson (1990):894-897; Merges & Nelson (1994):13-16.

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

[281] That is not to say that there have not been important patents that have allowed a single firm for a time to dominate a particular sector. The Rambus patent is the most common example mentioned in my interviews. Urdahl. But 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 a dominating 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).

[282] Bessen & Maskin (2002):5.

[283] I am working with John Allison on a project that will address that particular point more specifically by integrating financial data and patents obtained by firms in the Software 500.

[284] Bessen (2003).

too low, particularly in cases in which mature incumbents populate the industry.[285] 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. And it is clear that IBM has determined for reasons of its own – influenced to be sure by federal antitrust enforcement[286] – 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,[287] the historical events that have made IBM and now Microsoft as cautious as they are make it difficult to argue that those patterns have emerged,[288] and unlikely that those patterns will emerge in the foreseeable future.[289]

What all of this means for purposes of potential patent thickets is that potential innovators know that the large mass of existing patents held by IBM and Microsoft are likely to receive 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.[290] The fact that it can earn royalties from those patents

---

[285] Bessen (2003):14-15.

[286] *See supra* note 269.

[287] 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 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/.

[288] 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).

[289] The related paper (Bessen & Hunt (2003)) argues that patents have caused sub-optimal investment in the software industry. As with Bessen & Maskin (2002), the data set is so loosely connected to the industry as a whole as to make their conclusions inherently suspect. For present purposes, however, a few points are noteworthy. For one thing, they do not collect patents for the software industry. Rather, they collect patents that based on key words are reasonably likely to be patents that involve software. Many of those patents, as they recognize, are not by firms in the software industry at all. Also, their dataset involves only firms that are on CompuStat, which excludes the majority of firms in the software industry. In particular, to the extent that a significant part of investment in the industry is not included in CompuStat – it includes none of the venture-capital investment that this paper examines – their dataset cannot be taken to support any conclusion about the industry as a whole.

[290] *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

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 plausible 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 most unlikely based on the interviews that I have conducted. 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.

3. The Effects of Software Patents

The preceding sections are only a partial analysis of the effects of software patents. At best, they establish that the ready availability of technology through cross licensing minimizes the adverse effects of patents in the software industry. Thus, although individual firms surely find it valuable that they can use patents as barter in cross-licensing transactions, that arrangement does nothing to show that patents as a whole are beneficial in the software industry. After all, you might think it would be cheaper for everybody if we did not have patents if everybody in the industry is going to allow everybody else free access to technology.[291]

If the role of patents in cross licensing is to be positive, they will have to provide some benefit that exceeds the transaction costs of obtaining the patents. For example, it

---

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.

[291] Remember, cross licensing in the software industry as it currently exists is not a system for disseminating technology, it is simply a device for avoiding patent liability. Firms that cross-license do not generally help each other understand the licensed technology; they simply give free rein to develop products despite potential claims of patent infringement.

might be that it is easier to share technology cooperatively in a world with patents than in a world without it. 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. With patents, firms that wish to use their technology offensively must accept the likelihood of suit by other industry players for infringement of the patents held by those players. In an industry where innovation is so cumulative that all products include innovations first discovered by numerous players, and where it is difficult to produce a major product except in a company of substantial size, that could be an important consideration.

As a second possibility, patents might provide an effective way of evaluating the value of the technology that each firm has so as to determine the amount and direction of payment that is appropriate for each contracting pair. 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.

Of course, cross licensing is not the only patent benefit discussed above.[292] The analysis of signaling, codification of tacit knowledge, and aiding acquisitions suggests benefits that directly facilitate the flow of funds into the software firm and thus provide direct support for innovation in the industry.[293] Similarly, the ability to shelter small firms at an early stage in their competitive growth could have a significant effect on the structure of the industry. If the literature on innovation suggests that cumulative innovation proceeds better when more investigators attempt to resolve a particular problem, then an institution that supports the proliferation of firms should foster an industrial structure more conducive to innovation.[294] If – as I argue – the principal direct benefits of patents protect small firms and help them obtain financing, then software patents may play a role in the high levels of fractionation we observe in the industry as it currently exists.[295] It surely is true that patents are not as effective in appropriating the

---

[292] *See supra* Sections VI(B)(2)(b)-(d).

[293] It is not clear, however, that those same benefits could not be produced with a system that did not burden later innovators by excluding them from the covered technology.

[294] Merges (1996a) presents a thoughtful discussion about the role of IP in the structure of the software industry. He takes the view that IP protection should be set to accommodate the view "that having more, smaller firms is better than fewer, larger firms." P. 287. For similar sentiments, see Langlois & Steinmuller, in MOWERY & NELSON (1999):50, 55 (discussing the semiconductor industry).

[295] Writing in 1996, Merges discusses the possibility that stronger IP might foster innovation by facilitating entry of venture-backed firms that might develop "multiple, rivalrous technologies." Merges (1996a):287-90. He was not persuaded at that time that stronger IP would have that effect. In my view, the development of the industry in the intervening years supports the more optimistic of his predictions. More recently, he argues with Arora that stronger IP rights can facilitate joint production in industries in which suppliers and buyers are not vertically integrated. Arora & Merges (2001). That argument is consistent with the analysis here, which

value of software inventions as they are in appropriating the value of some other kinds of inventions. Thus, the protection that patents afford may be relatively thin. That surely is the implication of the empirical evidence discussed in the companion paper. Still, even a thin protection that has positive effects is a good thing.[296]

It is of course quite difficult to *prove* that the industry is better off with patents, because it is difficult to construct a factual experiment that involves the modern software industry without patents. The closest thing we to such an experiment is Petra Moser's recent work related to 19th century patent fairs.[297] That work demonstrates, at least in the context that she examines, that the absence of patent protection has a substantial effect on the allocation of resources. She shows that innovative resources in economies without patent protection flow to industries where other methods of appropriating the value of inventions make patent protection of relatively limited importance. At the same time, innovative resources flow away from industries where patents are an important method of appropriating value. If I am right in arguing above that patents provide significant assistance to small companies trying to appropriate the value of their software inventions, then, her work suggests, the removal of patent protection might lead to a flow of innovative resources away from the industry.

The shifting nature of IP protection in the industry makes it difficult to be sure of the effect patent protection currently has. It is true, for example, that the industry thrived in the early days of the 1970's and 1980's without broad patent protection. As suggested in Part II, however, copyright protection was broader then that it has become during the last decade. More fundamentally, the structure of the industry and its products was very different then. As the industry has matured, innovation has become more complicated and technical than anybody could have imagined in the 1960's. Thus, there has been a considerable increase both in the amount of investment required and in the benefits of having multiple firms working on cutting-edge problems. If patents are helping to direct venture-capital resources into firms in investment-intensive sectors of the industry, then it is plausible to think that there is more innovation in those sectors than there would be without patents.

Bessen's arguments that R&D spending has fallen during the years since software patents became available could weigh against my thesis, but the evidence does not provide much factual support for his argument. For one thing, his focus on large patentholders in a particular class – rather than the software industry – makes his dataset almost useless for this purpose. Most obviously, most of the largest patentholders in the particular sectors that he examines are large manufacturing firms (Ford, General Electric, and Japanese firms like Mitsubishi and Matsushita), for whom software patents are at best a minor part of the firm's operations. Yet the paper does not even attempt to

---

suggests that the IP rights can enhance the ability of numerous smaller firms attempting to develop technology that would be useful in products sold by a small number of large firms.

[296] For a similar line of reasoning, see Burk & Lemley (2003).

[297] Moser (2003).

determine the amount of R&D spending those firms actually allocate to software development. The idea that changes in American software patenting rules are altering the overall R&D spending of those firms seems so dubious as to render the statistical analysis of the data of no value.

Moreover, a glance at data that does address the software industry suggests quite a different picture. *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.[298] By comparison for the same period, it has been 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 quite high in the software industry in comparison to other industries.[299] Given the limited manufacturing materials that go into software, those high figures should not be surprising. But it is hard to credit Bessen's argument that R&D spending in the industry is systemically depressed.

National Science Foundation data regarding industrial R&D intensity provides a 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%.[300] 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).

As this data shows, software R&D spending tends to be relatively stable from year to year as a percentage of sales. Indeed, the most obviously significant variable in R&D spending within the industry appears to be company size. Relying on data from CompuStat, 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

---

[298] 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.

[299] 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. But it is plain that the relative levels of development have historical reasons that have little to do with patent protection. *See* CAMPBELL-KELLY (2003); Mowery (in MOWERY & NELSON (1999)).

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

years.[301] 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%[302]; 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%.[303]

For now, perhaps the most that can be said with clarity is the basic point that began this section: the patent system is not systematically preventing the initiation of product development. Beyond that, it is plain that the system is not obviously dysfunctional. In the world that we have – with patents – there are literally hundreds of small firms using institutional financing to develop new technologies. 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.

4. A Note on Open Source

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 the open-source movement. 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.[304] Those who work in that community may have little or no 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

---

[301] 2001: 16.5%, 26.9, 41.7; 2000: 15.9%, 28.0, 52.4; 1999: 15.1%, 20.0, 41.4. I note that the 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.

[302] Interestingly enough, 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).

[303] A similar trend is evidenced in the preceding three years. I expect to analyze the relation between patenting and R&D intensity in the Software 500 in the companion paper with John Allison.

[304] The literature on that subject is large and contentious. For a lucid, accessible, and reasonably balanced introduction, see FINK (2002).

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 the open-source community. To put the matter in a current context, suppose for a moment that the Linux operating system in fact does infringe in a substantial way patents held by SCO. That could[305] 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 have no patents of their own with which they might protect themselves in such litigation. At the same time, it 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 long run.

That raises the question, in turn, whether the potential for high-quality software development by the open-source movement justifies eradication of software patents even for the commercial software industry. It is impossible to answer that question definitively without evidence that would allow a comparative weighing of the benefits of open-source software against the benefits that the commercial software industry derives from IP. In any event, that question is far beyond the scope of this paper. I note it here only to define the bounds of my analysis. The primary goal of this paper is to shed light on the role of IP in the commercial software industry. I believe I have offered some good reasons to think that role might be positive. Whether that role justifies any harm IP might pose to open-source developers is a question left for another day.

VII. Conclusion

We know little about the process of innovation generally and about how innovation works in the software industry. This paper attempts to grapple with those questions by identifying a point of significance in understanding the role of IP in industries of cumulative innovation – such as the software industry in this country. In those industries, IP can play a crucial role in innovation. It is not, however, the old-economy Chandlerian[306] way of inducing massive investment in R&D by incumbent firms. On the contrary, it serves as a device to foster high degrees of fractionation. By increasing the number of approaches to resolving important problems, fractionation can provide for an ever more rapid pace of innovation and consequent Schumpeterian

[305] This assumes, of course, that SCO has not in some manner (through use of the General Public License or otherwise) lost its right to enforce those patents in this context.

[306] *See* Chandler (2001) (arguing that the key to success in cutting-edge industries is developing comprehensive competence in a single firm).

destruction.[307] From that perspective, IP can be an engine of competitive discipline, and not a tool to block competition.

[307] Mowery & Nelson (in MOWERY & NELSON (1999)):369-70 (explaining that continuous entry of new firms into the software industry limits the value of "dominant design" theories and "first mover" advantage in explaining the structure of the software industry). *See also* LIEBOWITZ & MARGOLIS (1999) (case studies discussing failure of once-dominant software firms).

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, *Empirical Evidence on the Validity of Litigated Patents*, 26 AIPLA QUARTERLY J. 185 (1998)

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, *The Growing Complexity of the United States Patent System*, 82 B.U.L. REV. 77 (2002)

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

John R. Allison & Emerson H. Tiller, *Internet Business Method Patents: Law, Policy and Politics* or *The Business Method Patent Myth* (Feb. 2003 draft)

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

Ashish Arora, Marco Ceccagnoli & Wesley M. Cohen, *R&D and the Patent Premium* (Jan. 2003) (NBER Working Paper 9431)

Ashish Arora & Robert P. Merges, *Property Rights, Firm Boundaries, and R&D Inputs* (2001 unpublished manuscript)

Jonathan M. Barnett, *Private Protection of Patentable Goods* (Fordham Univ. Sch. of Law Research Paper 28) (September 2003)

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)

Timothy F. Bresnahan & Shane Greenstein, *Technological Competition and the Structure of the Computer Industry*, 47 J. INDUS. ECON. 1 (1999)

Stephen Breyer, *The Uneasy Case for Copyright: A Study of Copyright in Books, Photocopies, and Computer Programs*, 84 HARV. L. REV. 281 (1970)

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

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

Dan L. Burk & Mark A. Lemley, *Is Patent Law Technology-Specific*, 17 BERKELEY TECH. L.J. 1155 (2002)

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)

Christopher S. Cantzler, State Street*: Leading the Way to Consistency for Patentability of Computer Software*, 71 U. COLO. L. REV. 423 (2000)

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

Rochelle Cooper Dreyfuss, *Are Business Method Patents Bad for Business?*, 16 SANTA CLARA COMP. & HIGH TECH. L.J. 263 (2000)

Rebecca S. Eisenberg, *Analyze This: A Law and Economics Agenda for the Patent System*, 53 VAND. L. REV. 2081 (2000)

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)

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

Ron Gilson, *Engineering a Venture Capital Market: Lessons from the American Experience* (Stanford Law School, John M. Olin Program in Law and Economics, Working Paper 248) (November 2002)

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

Paul A. Gompers & Josh Lerner, *The Determinants of Corporate Venture Capital Success: Organizational Structure, Incentives, and Complementarities* in CONCENTRATED CORPORATE OWNERSHIP (Randall K. Morck ed. 2000)

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

Mark F. Grady & Jay I. Alexander, *Patent Law and Rent Dissipation*, 78 VA. L. REV. 305 (1992)

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, *Business Method Patents, Innovation, and Policy* (May 2003 draft available at www.ssrn.com)

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

Mark A. Haynes, *Black Holes of Innovation in the Software Arts*, 14 BERKELEY TECH. L.J. 567 (1999)

Herbert Hovenkamp, Mark Janis & Mark A. Lemley, *Anticompetitive Settlement of Intellectual Property Disputes*, 87 MINN. L. REV. 1719 (2003)

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

Brian Kahin, *Policy Development for Information Patents in the U.S. and Europe* (June 3, 2002 draft)

Steven N. Kaplan, Berk A. Sensoy & Per Strömberg, *How Well Do Venture Capital Databases Reflect Actual Investments* (9/2002 draft)

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

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

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

F. Scott Kieff, *Property Rights and Property Rules for Commercializing Inventions*, 85 MINN. L. REV. 697 (2001)

Edmund W. Kitch, *The Nature and Function of the Patent System*, 20 J.L. & ECON. 265 (1977)

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, *Assessing the Contribution of Venture Capital to Innovation*, 31 RAND J. ECON. 674 (2000)

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

Richard N. Langlois & W. Edward Steinmueller, *The Evolution of Comparative Advantage in the Worldwide Semiconductor Industry, 1947-1966*, in MOWERY & NELSON (1999)

Jean O. Lanjouw, Ariel Pakes & Jonathan Putnam, *How to Count Patents and Value Intellectual Property: The Uses of Patent Renewal and Application Data*, 46 J. INDUS. ECON. 405 (1998)

Jean O. Lanjouw & Mark Schankerman, *Stylized Facts of Patent Litigation: Value, Scope and Ownership* (Suntory and Toyota International Centres for Economics and Related Disciplines, LSE) (Discussion Paper No. EI/20, January 1998)

Jean O. Lanjouw & Mark Schankerman, *Characteristics of Patent Litigation: A Window on Competition*, 32 RAND J. ECON. 129 (2001)

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

Mark A. Lemley, *Reconceiving Patents in the Age of Venture Capital*, 4 J. SMALL & EMERGING BUS. L. 137 (2000)

Mark A. Lemley, *Rational Ignorance at the Patent Office*, 95 NW. U. L. REV. 1495 (2001)

Mark A. Lemley, *Intellectual Property Rights and Standard-Setting Organizations*, 90 CAL. L. REV. 1889 (2002)

Mark A. Lemley & Colleen V. Chien, *Are the U.S. Patent Priority Rules Really Necessary?*, 54 HASTINGS L.J. 1299 (2003)

Mark A. Lemley & Ragesh K. Tangri, *Ending Patent Law's Willfulness Game* (unpublished 2003 manuscript)

Mark A. Lemley & David W. O'Brien, *Encouraging Software Reuse*, 49 STANFORD L. REV. 255 (1997)

Josh Lerner, *Patenting in the Shadow of Competitors*, 38 J.L. & ECON. 463 (1995)

Josh Lerner and Jean Tirole, *Some Simple Economics of Open Source*, 50 J. INDUS. ECON. 197 (2002)

Josh Lerner & Jean Tirole, *The Scope of Open Source Licensing* (Nov. 19, 2002)

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

Jonathan Levin and Richard Levin, *Patent Oppositions* (Oct. 2002 draft)

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 LIEBOWITZ, RE-THINKING THE NETWORK ECONOMY (2002)

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)

Richard S. Markovits, *A Basic Structure for Microeconomic Policy Analysis in Our Worse-Than-Second-Best World: A Proposal and Related Critique of the Chicago Approach to the Study of Law and Economics*, 1975 WIS. L. REV. 950

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, *Tailoring Legal Protection for Computer Software*, 39 STANFORD L. REV. 1329 (1987)

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, *Who Owns the Charles River Bridge? Intellectual Property and Competition in the Software Industry* (4.2.99 draft)

Robert P. Merges, *As Many As Six Impossible Patents Before Breakfast: Property Rights for Business Concepts and Patent System Reform*, 14 BERKELEY TECH. L.J. 577 (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. (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)

Micro-and Macroeconomic Implications of the Patentability of Software Innovations, Intellectual Property Rights in Information Technologies Between Competition and Innovation (Sept. 2001) (prepared for German Federal Ministry of Economics and Technology by Fraunhofer-Institute for Systems and Innovation Research and Max-Planck Institute for Foreign and International Patent, Copyright and Competition Law)

JOEL MOKYR, THE GIFTS OF ATHENA: HISTORICAL ORIGINS OF THE KNOWLEDGE ECONOMY (2002)

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)

RICHARD R. NELSON & SIDNEY G. WINTER, AN EVOLUTIONARY THEORY OF ECONOMIC CHANGE (1982)

Francesco Parisi, *Entropy in Property*, 50 AM. J. COMP. L. 595 (2002)

Francesco Parisi, *Freedom of Contract and the Laws of Entropy* (unpublished manuscript)

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

Francesco Parisi & Catherine Ševčenko, *Lessons from the Anticommons: The Economics of* New York Times Co. v. Tasini, 90 KENTUCKY L.J. 295 (2001-02)

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

Arti Kaur Rai, *Regulating Scientific Research: Intellectual Property Rights and the Norms of Science*, 94 NW. U. L. REV. 77 (1999)

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

David A. Rice, *Public Goods, Private Contract and Public Policy: Federal Preemption of Software License Prohibitions Against Reverse Engineering*, 53 U. PITT. L. REV. 543 (1992)

David A. Rice, Sega *and Beyond: A Beacon for Fair Use Analysis . . . At Least As Far As It Goes*, 19 U. DAYTON L. REV. 1131 (1994)

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)

Carol Rose, *The Comedy of the Commons: Custom, Commerce, and Inherently Public Property*, 53 U. CHI. L. REV. 711 (1986)

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)

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

Norbert Schulz, Francesco Parisi & Ben Depoorter, *Fragmentation in Property: Towards a General Model* (Würzburg Economic Papers Nr. 32) (Nov. 2001)

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

Puay Tang, John Adams & Daniel Paré, *Patent Protection of Computer Programmes* (2001) (prepared for the European Commission, Directorate-General Enterprise)

Nidhi Thakur, *The Recent Surge in Software Patenting in the United States: An Empirical Analysis of Pure Software Firms* (2003 draft)

George G. Triantis, *The Role of Debt in Interactive Corporate Governance*, 83 CAL. L. REV. 1073 (1995)

George G. Triantis, *Debt Financing and Motivation*, 31 U. RICHMOND L. REV. 1323 (1997)

George G. Triantis, *Financial Contract Design in the World of Venture Capital*, 68 U. CHI. L. REV. 305 (2001) (reviewing GOMPERS & LERNER)

ERIC VON HIPPEL, THE SOURCES OF INNOVATION (1988)

R. Polk Wagner, *Information Wants To Be Free: Intellectual Property and the Mythologies of Control*, 103 COLUM. L. REV. 995 (2003)

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

Harold R. Weinberg & William J. Woodward, Jr., *Legislative Process and Commercial Law: Lessons from the Copyright Act of 1976 and the Uniform Commercial Code*, 48 BUS. LAWY. 437 (1993)

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