

The Software Patent Experiment[†]

James Bessen*
Research on Innovation and Boston University (visiting)

Robert M. Hunt**
Federal Reserve Bank of Philadelphia

March 16, 2004

* jbessen@researchoninnovation.org

** Ten Independence Mall, Philadelphia, PA 19106. Phone: (215) 574-3806. Email:
bob.hunt@phil.frb.org

Thanks to Peter Bessen of May8Software for providing a software agent to acquire our patent database and Annette Fratantaro for her work with the Compustat data set. Also thanks to John Allison, Tony Breitzman and CHI Research, Iain Cockburn, Dan Elfenbein, Bronwyn Hall, Joachim Henkel, Brian Kahin, Cecil Quillen, Leonard Nakamura, Eric von Hippel, Rosemarie Ziedonis and seminar participants at APPAM, Berkeley, EPIP Munich, Federal Reserve Banks of Philadelphia and San Francisco, the Federal Reserve System Applied Micro meetings, Harvard, IDEI, MIT, NBER, and OECD.

[†] The views expressed here are those of the authors and do not necessarily represent the views of the Federal Reserve Bank of Philadelphia or the Federal Reserve System.

INTRODUCTION

When it comes to patents, over the last two decades, the U.S. has undergone an almost accidental process of legal innovation. Standards were reduced—we now issue patents for inventions that, in the past, would not have qualified for protection. In addition, the scope of technologies that can be patented have been increased to include, among other things, gene sequences, computer programs, and methods of doing business.

This article investigates the effects of extending the patent system to a field of technology—computer software—known for rapid innovation well before software patents became commonplace. According to our estimates, the United States Patent and Trademark Office (USPTO) now grants at least 20,000 software patents a year and the numbers are growing rapidly. The European Commission is considering a proposal to formally recognize the patentability of computer programs in member countries. These changes have been controversial, but they are typically justified by the argument that making patents easier to obtain will increase the incentive to invest in R&D and therefore the rate of innovation.¹ In policy circles, it is fair to say this is the conventional wisdom.

There is sound empirical evidence that, for at least some industries, the availability of patents is an important factor that explains the willingness of firms to invest in R&D. For example, a number of surveys establish the important role that patents play in the U.S. chemical and pharmaceutical industries.² But these surveys also show that in many other industries patents are not regarded as either very important or effective in protecting one's innovations. Other general reviews of the effects of the patent system reach ambiguous conclusions—patents help in many circumstances, but not in others, and in some instances the effects may be deleterious.³

¹ Many studies examine the relationship between growth in R&D and growth in productivity or economic output. See, for example, Guellec and van Pottelsberghe de la Potterie (2001) and Griliches (1994).

² See Richard Levin et al (1987) and Cohen et al (2000). Using data from the latter survey, Arora et al (2003) find that firms who rate patents as both more important and more effective tend to do more R&D.

³ For recent reviews, see Federal Trade Commission (2003), National Research Council (2003), and Gallini

The research described in this article suggests there is some reason for concern about the economic effects of software patents.⁴ We find that software patents are not closely related to the creation of computer programs—the vast majority of software patents are obtained by firms outside of the software industry.⁵ We also find that firms that focus on software patents, in the sense that a higher share of their new patents are software patents, have tended to focus less on research than other firms. Interpreting these facts is difficult, but they do suggest that the relationship between the increased availability of software patents and the incentive to invest in research and development (R&D) is more complicated than is often assumed in the policy debate. In short, we did not find much evidence in favor of the conventional wisdom.

CHANGES TO OUR PATENT SYSTEM

The American patent system has changed in a number of important ways over the last quarter century. In this article, we explore the potential effects of some of these changes: the relaxation of standards used to determine whether an invention qualifies for patent protection and the elimination of the so-called subject matter exception that precluded the patenting of computer programs.

What is a Patent? For over 200 years, the U.S. government has used patents to reward inventors for their discoveries. The reward is a grant of the legal right to exclude others from making, using, or selling the patented invention for a limited period of time.⁶ If the patent is infringed, the patent owner may sue the infringer to recover lost profits. If the infringement was willful, additional damages may be awarded.

(2002).

⁴ This article is based on our 2004 working paper.

⁵ We identify software firms as those companies included in Standard Industry Classification (SIC) 7372 (software publishers) as coded in Standard and Poor's *Compustat* database in 1999. For some purposes, we use a broader definition of software firms, i.e., those classified in SIC 737 (computer programming, data processing, and other computer related services).

⁶ Today, a U.S. patent expires 20 years after the date of application. In the past, the patent term ran for 17 years from the date of grant.

In certain ways, a patent is a custom design. The inventor's right to exclude is limited to those *claims* applied for and granted by the patent office. Those claims are based, at least in part, on the description of the invention contained in the application to patent office.

Not Every Invention Can be Patented. U.S. patent law permits a patent to be granted only for inventions that are useful, new, and non-obvious. The first two requirements are fairly intuitive and sensible. One view of patents is that they are a bargain with inventors—the government grants a temporary monopoly on an invention, but only if it is both useful and represents an advance over our existing knowledge, which patent lawyers call the prior art. In exchange, the inventor must disclose the nature of the invention, which is described in the patent document itself. The third requirement, *nonobviousness*, is less clear. It rules out the patenting of an invention that would have been obvious to a practitioner in the relevant field at the time the invention was made. In other words, a patentable invention must be more than a trivial extension of the prior art.

Our patent law, and many judicial decisions, provides instructions on how the nonobviousness requirement is to be applied. During the 1980s, a number of judicial decisions revised these instructions in significant ways. In practice, the modified test for nonobviousness is easier to satisfy than the one applied prior to the early 1980s. As a result many more inventions now qualify for patent protection.⁷ Other judicial decisions made it easier for a patent holder to obtain a preliminary injunction—a court order prohibiting a potentially infringing activity even before the question of infringement is decided by the court. Today, the threat of a preliminary injunction often carries significant weight in negotiations between patent holders and alleged infringers.

Subject Matter Exceptions. As a general principle, the American patent system is not designed to treat different kinds of inventions differently. For example, when

⁷ The changes in the 1980s were instituted by the Federal Circuit, a specialized appeals court for patent and certain other cases, created in 1982. For more information about these decisions, and their affect on subsequent litigation, see Hunt (1999) and Jaffe (2000). The economic effects of reduced patentability standards are examined in Hunt (forthcoming).

Congress passed the 1952 Patent Act, the committee report endorsing the bill stated that the new law was meant to apply to “everything under the sun made by man.”⁸ These words are often mentioned in judicial decisions where a federal court is confronted with the problem of interpreting Congress’ intent in drafting that law.

One exception to this rule was computer software. In the 1972 decision *Gottschalk v. Benson*, the Supreme Court ruled that the computer program in question was a mathematical algorithm and, as such, was unpatentable subject matter. But it did not take very long before new decisions began to blur this seemingly bright line distinction between computer programs and other inventions. For example, in the 1981 decision in *Diamond v. Diehr*, the Supreme Court ruled that an invention incorporating a computer program could be patented as long as the new and non-obvious aspects of the invention did not consist entirely in the software. Even this distinction was gradually eroded.⁹

Any real difference in the treatment of software and other inventions was essentially eliminated after a 1994 appeals court decision (*in re Alappat*) upheld the patentability of a computer program that smoothes digital data before displaying it as a waveform on a computer monitor. Shortly after that decision, the Patent and Trademark Office issued a comprehensive revision to examination guidelines for computer related inventions. Thereafter, the number of software patents granted increased dramatically (see *Software Patents Granted*).

TAKING A CLOSER LOOK AT SOFTWARE PATENTS

Despite considerable interest in the effects of granting patents on computer programs, there is no official list of software patents. USPTO maintains a detailed system for classifying patented inventions by technology field—a sort of Dewey decimal system for patents. But there is no explicit classification for software inventions. Instead, researchers

⁸ Senate Report No. 1979 82d Congress, 2nd Session (1952), p. 5.

⁹ For additional information on the changing treatment of software in patent law, see Hunt (2001).

must devise their own ways of identifying software patents.¹⁰ The data used in this article are based on a simple keyword search of USPTO's database of patents issued after 1975. We looked for patents that used the words "software" or "computer program" in the description of the invention.¹¹

According to this definition, about 1,000 software patents a year were granted in the early 1980s, increasing to about 5,000 a year in 1990. The rate doubled again by the by 1996. Nearly 25,000 software patents were granted in 2002. This was a period of very rapid growth in patenting—the number of patents of any kind granted in 2001 was 1.7 times larger than in 1981—but the growth in software patents was much larger still. As a result, the share of all patents that are counted as software patents increased from about 2% in the early 1980s to nearly 15% by 2002 (See *Software Patent Share*).

Software patents are an American phenomenon. We can learn something about the inventors and owners of patents by examining information contained on the patent document itself. From this information, we learn that software patents are a (relatively) home grown phenomenon. During the 1990s, 70% of software patents were obtained by inventors living in the U.S.; that is significantly higher than the 53% domestic share of inventors of all other patents. Similarly, 70% of all software patents owned by companies went to firms headquartered in the U.S.; 51% of all other patents owned by companies went to an American firm.

Established firms obtain most software patents. The typical owner of a software patent is a relatively large, well-established firm. During the 1990s companies obtained a larger share of software patents than other patents (88% vs. 80%).¹² To put it another way, individuals were relatively less likely to obtain their own software patent

¹⁰ For details on the different approaches see Allison and Lemley (2000), Allison and Tiller (2001), and Graham and Mowery (2003).

¹¹ The exact search query is found in the Data Appendix. For a comparison of this definition, and the resulting patent counts, with others in the literature, see our working paper.

¹² The USPTO data indicates the owner of a patent at the time it was issued. The owner may be the individual (s) who made the invention or an organization (assignee) such as firm or a government agency.

than a patent on another kind of invention.

We can also compare the financial characteristics of firms that obtain software patents and other kinds of patents.¹³ To do this, we examine the median firm ranked in terms of the number of (1) software patents obtained and (2) the number of other patents obtained.¹⁴ The median firm ranked in terms of software patents is much larger than the median firm ranked by other patents. If size is measured in terms of market value, the median software patentee is twice as large as the median patentee of other inventions (\$24 million vs. \$12 million). Measured in terms of sales, it is 50% larger (\$13 million vs. \$9 million). Measured in terms of spending on research and development, it is 68% larger (\$956 million vs. \$376 million).

Most software patents don't come from the software industry. We were surprised to find that the vast majority of software patents are not obtained by firms associated with computer software. In the second half of the 1990s, firms in the software industry received 1% of all patents granted to firms that are included in the Compustat file and at most 7% of all software patents (see Table 1).¹⁵ Manufacturers accounted for 3 out of 4 software patents. The top 5 firms in terms of software patents obtained in 1995 included IBM, Motorola, Hitachi, AT&T, and Hewlett-Packard. Nine of the top 10 firms ranked by software patents received in 1995 were on the list of the top 20 firms ranked by patents of any kind.

Firms in just three manufacturing industries (machinery, electronics, and

¹³ Much of our analysis is based on matching patents to companies in Standard and Poor's Compustat database. We rely extensively on data contained in the NBER Patent Citations Data File and data generously provided to us by Tony Breitzman of CHI Research. For details on the matching process see the Data Appendix.

¹⁴ The median identifies the firm where 50% of all firms have more patents than it does and 50% of all firms have fewer patents than it does.

¹⁵ These statistics are for successful patents applied for during 1994-7. For this calculation, the software industry is defined as firms included in the SIC 737, but excluding IBM which alone accounted for 6% of software patents granted.

instruments) alone accounted for two thirds of software patents granted to firms—a number that significantly exceeds their impressive 54% share of patents of any kind. These numbers are even more remarkable when we examine the distribution of computer programmers across these industries.¹⁶ These are presumably the workers responsible for creating most new computer programs. Manufacturers of machinery, electronics, and instruments employed only 6% of all computer programmers and yet they obtained 2 out of 3 software patents. Firms outside the manufacturing sector employed 9 out of 10 computer programmers; but together they accounted for only 1 out of 4 software patents. It would appear that the distribution of software patents across industries reflects something other than the creation of software.

In the next two sections we describe a number of findings from our statistical analyses. First, we explore differences in the software patenting behavior of firms across industries and over time. Then, we look for any relationships between a firm's software patenting behavior and its R&D investments.

WHY ARE THERE SO MANY SOFTWARE PATENTS?

The most obvious answer to this question is that firms obtain software patents today, when they either could not or did not obtain them in the past. In addition, the software sector of the economy has grown over time. But this explanation tell us very little and it potentially exaggerates the effects of legal changes by ignoring economic and other factors that may have contributed to the explosion in software patenting. We will try to control for some of these factors and then examine the remaining variation in software patenting behavior across firms and over time.

Estimating Patent Propensity. Industries vary significantly in their *propensity to patent*—that is, the average number of patents obtained from a given amount of resources spent on developing new products and processes. For example, during the mid 1990s firms in the machinery, electronics, and instrument industries received between 4 and 7

¹⁶ Our data on computer programmers comes from various editions of the *Occupational Employment Survey*, published by the Bureau of Labor Statistics. We thank Joseph Bush of BLS for his assistance.

patents (of any kind) for every \$10 million in R&D they spent (see *Distribution of Software Patents*). That compares to only about 1 patent per \$10 million in R&D for firms in the software industry. Based on this simple calculation, all else equal, if we observed a \$10 million increase in R&D in each of these industries, we would expect to see 4-7 more patents by manufacturers of machinery, electronics, and instruments and 1 additional patent by software companies.

Our statistical analysis attempts to explain the number of software patents that firms obtain. Our technique is a more complicated version of the calculation described above. We control for a firm's investment in R&D and plant and equipment, its size, and other characteristics. We also control for employment of computer programmers and engineers in the firm's industry. We say that a firm, or an industry, has a higher software patent propensity if, it generates more software patents after controlling for R&D and these other factors. We also look for changes in software patent propensity over time, that is changes in the number of software patents obtained by firms that cannot be explained by changes in their R&D or other factors.¹⁷ This residual change in patent propensity over time should capture, albeit imperfectly, the effect of changes in the legal treatment of software patents.

What Explains Software Patent Propensity? We find that firms apply for more software patents when they are both more research and capital intensive and when the industry workforce consists of more programmers and engineers. We did not find a difference in the software patent propensity of old and young firms, except in the software industry. There, new firms have a significantly lower software patent propensity than older firms in the same industry.¹⁸

Manufacturers in general have a much higher software patent propensity than do firms in the software industry (see *Distribution of Software Patents*). After controlling for

¹⁷ For an excellent example of this methodology applied to the American semiconductor industry, see Hall and Ziedonis (2001).

¹⁸ A new firm in our analysis represents the first five years that a company is reported in Compustat.

R&D and other factors, firms in the machinery, electronics, and instruments industries obtain software patents at a rate 4 to 10 times higher than firms in the software sector. In addition, software patent propensity is significantly higher for firms in industries where their peers obtain more patents (of all kinds) per employee. These patterns suggest that a firm's acquisition of software patents is better explained by its industry's reliance on patents in general than by its investments in the development of software.

The Rise in Software Patent Propensity Over Time. The average annual increase in the number of successful applications for software patents over the years 1987 to 1996 was 16%. Our analysis shows that changes in firms R&D and capital investments, employment of programmers, and other factors explains about one-third of the rate of increase in software patents. The remaining two-thirds (about 11% a year) represent the increase in software patent propensity over time. Compared to the rate for 1987, and holding all other factors constant, firms were successfully applying for 50% more software patents in 1991, and more than 150% more by 1996.

It is likely that a good part of this increase in patent propensity is the result of changes in the legal treatment of software patents.¹⁹ Such changes might work in two ways. The cost of obtaining software patents, relative to the cost of obtaining any other patent may have fallen. Alternatively, the economic benefit conferred by a typical software patent, again relative to the benefit conferred by any other kind of patent, may have increased. Or both of these may be true—in other words, our analysis suggests that the relative profitability conferred by obtaining software patents increased over time.

DO MORE SOFTWARE PATENTS MEAN MORE R&D?

Ordinarily when a firm is able to obtain additional patents, the profits earned on its

¹⁹ We cannot attribute all of this residual increase to legal changes, because the pattern can also be explained by productivity growth, i.e. increases in the number of inventions per programmer. Our review of measures of productivity growth in the software sector suggests, very roughly, that an upper bound of productivity growth is unlikely to explain more than half of the residual increase in software patent propensity over time.

inventions should rise. This should encourage the firm to do more R&D. Similarly, when a firm performs more R&D, it should invent more, and that should make it easier to get additional patents. This is the traditional incentive theory of patents: By granting firms more and stronger property rights—that is, the right to capture more profits from their R&D investments—the government can stimulate innovation.

Complements or Substitutes? When economists think about a problem like this, they often inquire whether the variables in question are *complements* or *substitutes*. In the standard textbook exposition, two goods are complements if a *fall* in the price of one good induces an *increase* in the consumption of the other. An example of two goods that might act as complements might be coffee and cream. On the other hand, two goods are substitutes when a *fall* in the price of one good causes a *decrease* in the consumption of another good. An example of two goods that might act as substitutes would be public transit and automobiles.

This intuition about complements and substitutes also applies to a firm's choice of inputs used by a firm. For example, if the cost of information technology (IT) declines, it is entirely possible that a firm will purchase more of the technology *and* hire more computer programmers who are skilled in using that technology. If that did happen, we would say that IT equipment and computer programmers are complementary inputs.²⁰ If on the other hand, we observed a decline in the number of computer programmers, we might conclude that IT equipment has substituted for computer programmers, who have become more expensive relative to cost of IT equipment.

To economists, then, the conjecture that making software patents easier to obtain will increase investments in R&D is a claim that these patents and R&D are complementary inputs in the production of profitable innovations. All else equal, the legal changes described earlier increased the return from obtaining software patents relative to other patents. We have already seen that one response was a very large

²⁰ Note that the total value of IT equipment purchased might rise or fall depending on whether or not the firm's demand for IT is elastic. We say demand is elastic when a decline in price, in percentage terms, induces a larger increase, again in percentage terms, in the quantity demanded.

increase in the number of such patents (the quantity demanded rose as the cost of software patents fell), even after controlling for other factors. But what has been the effect on demand for R&D?

The Relationship Between R&D and Software Patents Has Changed. To answer this question, we might examine the relationship between changes in a firm's research intensity (typically measured by the ratio of R&D to sales) and changes in the firm's *software patent focus*—their new software patents divided by all their new patents—over time. An increasing software patent focus should reflect a decline in the firm's cost of obtaining software patents relative to other patents. A positive correlation between changes in research intensity and changes in software patent focus would suggest that software patents and R&D are indeed complementary inputs.²¹

Our actual statistical analysis is a more complicated version of this exercise. We examine differences in firms' R&D intensity and software patent focus over 5 year intervals. Of course many other factors might induce changes in R&D intensity, so we control for changes in input prices (including information technology), firm size, new vs. established firms, leverage, employment of computer programmers, and idiosyncratic factors specific to the firm or industry.²²

We find that, all else equal, during the late 1980s, firms that increased their focus on software patents tended increase their R&D intensity, but the relationship was weak. In other words, more likely than not, software patents and R&D were complementary inputs during the 1980s.²³ For the 1990s we found a much stronger relationship, but it was negative—all else equal, increases in software patent share were associated with

²¹ By positive correlation, we mean increases (decreases) in software patent focus are associated with increases (decreases) in R&D intensity in a way that cannot be explained by chance.

²² We used input prices data from the BLS Multifactor Productivity Series. Our regressions are based on cost shares or revenue shares (R&D/total costs and R&D/sales, respectively) to control for shifts in final demand. Examples of how economists use such techniques can be found in the Berndt (1991) and Greene (1993).

²³ But we cannot say with 95% confidence that this is true.

decreases in research intensity. This suggests that in the 1990s, software patents substituted for R&D. There is also evidence this negative correlation became even more negative by the late 1990s, but we cannot be 95% certain this is true.

The exact magnitude of the effect varies across firms distinguished by either their size or age, but not so much so that we can be sure the differences are not the result of chance. On the other hand, the effect is concentrated among firms in the machinery (including computers) and electronics (including semiconductors) industries and the software industry broadly defined²⁴—in other words, among firms in those industries that account for about two thirds of all software patents. Outside of those industries there is no systematic relationship between an increase in software patent focus and changes in firms' R&D intensity.

Overall, the effect is economically significant. Taking the analysis literally, if the number of software patents grew only as rapidly as all other patents after 1991, the average R&D intensity of U.S. firms would be about 7% higher than was actually recorded in 1997. This represents about \$9 billion in additional private R&D spending for the entire U.S. economy. It also represents about 5 years of the annual average increase in the research intensity of American firms since 1953.

But it is important to emphasize that the analysis does not identify the exact relationship that explains why an increases in software patent focus is associated with a decline in research intensity. In the language of statistics, this approach identifies a correlation but not causation. Still, we can compare the patterns identified in the data with a number of hypotheses about the effects of software patents.

RECONCILING THEORIES WITH THE DATA

While we can't provide a full explanation of what happened, we can compare the facts identified so far with a variety of hypotheses or theoretical arguments that appear in the

²⁴ Here we count all firms in SIC 73 (Business Services) as software and this includes IBM. If we exclude IBM from SIC 73, we do not find a systematic relationship between increases in software patent focus and changes in R&D intensity among the remaining firms in SIC 73.

debate over changes to the U.S. patent system.

The Incentive Theory. The first of these is the traditional incentive theory, which argues that by making available stronger property rights at lower cost, firms will have an increased incentive to engage in R&D. It is this conventional wisdom that is often cited in arguments in favor of extending patent protection to computer programs. Is our evidence consistent with this theory?

The answer seems to be no. We observe that the vast majority of software patents are obtained by firms outside the software industry and with little investment in the inputs (computer programmers) required to develop software inventions. The distribution of software patents seems to follow more closely the general pattern of industry-wide patent propensities than anything peculiar to software in itself. Industries known for prodigious patenting in general account for the vast majority of software patents obtained. And firms located in such industries have a higher software patent propensity.

The increases in the total number of software patents and in software patent share are consistent with firms responding to a decline the relative cost of obtaining these patents, or alternatively an increase in their cost effectiveness. But the negative correlation between increases in a firms' software patent focus and their R&D intensity suggest that firms may be substituting for R&D with software patents. This negative relationship is occurs in only certain industries (including computers and electronics), which just happen to be industries recognized for their prodigious patenting and the importance placed upon the strategic use of patents.²⁵

A Productivity Shock. Another hypothesis is that the U.S. economy has experienced a large productivity shock that favored inventions implemented via computer programs. Such a shock would be consistent with a large increase in software patenting and the long run trend towards higher research intensity among American firms. But it is inconsistent with the negative correlation between increases in software patent share and

²⁵ For evidence on the patenting behavior of such firms see Hall (2003). For a discussion of patent strategies in these industries see Levin et al (1987) and Cohen et al (2000).

R&D intensity. What's more, the observed increase in software patent propensity seems too large to explain entirely by advances in the productivity of computer programming.

One variation on the productivity shock hypothesis points to the potential for outsourcing of software development as the market for prepackaged software expanded. In other words, firms might have chosen to purchase software rather than to develop it internally. Such outsourcing could explain a decline in research intensity, but firms outsourcing their software development firm would also likely reduce their focus on software patenting. Conversely, software developers that benefit from outsourcing might be expected to increase both their R&D investments and software patenting. Neither of these patterns is consistent with the data.

It has also been suggested that the use of software in the R&D process significantly reduce the cost of doing R&D and this might explain the observed negative correlation between software patent focus and R&D intensity. But previous studies have shown that firms respond elastically to changes in the cost of doing R&D.²⁶ In other words the quantity of R&D firms do increases by at least as much in percentage terms, as the decrease in its cost. Thus, even if software reduced the cost of research and development, R&D intensity should not fall and it might even increase.

Patent Thickets? In contrast to the incentive theory already described, suppose instead that firms in an industry firms assemble large patent portfolios in order to extract royalties from competitors and to defend themselves from similar behavior by their rivals Economists have come to describe such an environment as a patent thicket.²⁷ In theory at least, extensive competition in patents, rather than inventions, may occur if firms rely on similar technologies and the cost of assembling large portfolios is not very high. In such an environment, firms may compete to tax each others' inventions and in the process

²⁶ See, for example, Berger (1993) and Hall and van Reenan (1999). In addition, our analysis takes into account changes in the cost of software and the share of computer programmers in the industry workforce.

²⁷ For evidence of this phenomena in the electronics and semiconductor industries see Grindley and Teece (1997) and Hall and Ziedonis (2001).

reduce their competitors' incentive to engage in R&D.²⁸ The outcome of patent litigation and licensing agreements often depends on the size of the firm's patent portfolio. This creates an incentive to build larger patent portfolios, especially when the firm focuses on patents as a competitive strategy. In this account, firms choose to compete in court, rather than in the marketplace.

The changing legal treatment of software patents might explain a systematic change in the behavior of some firms. During the early 1980s, patents were relatively costly to obtain, and this might have discouraged substitution away from R&D and toward strategic patenting. By the mid 1990s, software patents became a relatively inexpensive way to expand patent portfolios. This may have increased the attractiveness of a strategy that emphasizes patent rights over a strategy based on R&D. Such a change in strategy would be particularly attractive to mature firms if their R&D labs are not as productive as they once were.

The patent thicket explanation is consistent with the observed rise in patent propensity and the negative relationship between changes in software patent share and research intensity in certain industries. It might also explain why software patents are more common in industries with high patent propensities (machinery, electronics, and instruments) rather than in industries that focus primarily on developing software. And it is consistent with the observation that software patent propensity is higher in industries where firms obtain more patents per employee.

CONCLUSION

Nearly 50 years ago, scholar Fritz Machlup presented the results of his study on the efficacy of the patent system to the U.S. Congress. He concluded:

“If one does not know whether a system "as a whole" (in contrast to certain features of it) is good or bad, the safest "policy conclusion" is to "muddle through" If we did not have a patent system, it would be irresponsible... to recommend instituting one. But since we have had a

²⁸ For a theoretical model of this intuition see Bessen (2003).

patent system for a long time, it would be irresponsible, on the basis of our present knowledge, to recommend abolishing it.”

What would Machlup think about a significant expansion of the patent system, and a significant change in patentability standards, instituted in the absence of much evidence about the likely effects? Yet this is precisely what has happened in the U.S. over the last quarter century.

These changes are often justified on the basis of the conventional wisdom—that granting more and stronger property rights will necessarily stimulate innovation. Our evidence suggests this assumption may be incorrect in the case of software patents in the U.S. during the 1990s. If, instead, the legal changes have encouraged strategic patenting, then the end result might be less innovation. Clearly, much more research should be done on this important policy question.

Bibliography

- Allison, John R. and Mark A. Lemley. 2000. "Who's Patenting What? An Empirical Exploration of Patent Prosecution," *Vanderbilt Law Review*, Vol. 58, pp. 2099-2148.
- Allison, John R. and Emerson H. Tiller. 2001. "An Empirical Analysis of Internet Business Method Patents," mimeo, University of Texas at Austin.
- Ashish Arora, Marco Ceccagnoli, and Wesley M. Cohen. 2003. "R&D and the Patent Premium," NBER Working Paper No. 9431.
- Berger, Philip. 1993. "Explicit and Implicit Effects of the R&D Tax Credit," *Journal of Accounting Research*, Vol. 31, pp. 131-71.
- Bessen, James. 2003. "Patent Thickets: Strategic Patenting of Complex Technologies," ROI Working Paper.
- Bessen, James and Robert Hunt. 2004. "An Empirical Look at Software Patents," Federal Reserve Bank of Philadelphia Working Paper 03-17R, March 2004.
- Berndt, Ernest R. 1991. *The Practice of Econometrics: Classic and Contemporary*. New York: Addison-Wesley Company.
- Cohen, Wesley M., Richard R. Nelson, and John P. Walsh. 2000. "Protecting Their Intellectual Assets: Appropriability Conditions and Why U.S. Manufacturing Firms Patent (or Not)," NBER Working Paper No. 7552.
- Federal Trade Commission. 2003. *To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy*. Washington, DC: Federal Trade Commission.
- Gallini, Nancy T. 2002. "The Economics of Patents: Lessons from Recent U.S. Patent Reform," *Journal of Economic Perspectives*, Vol. 16, pp. 131-54.
- Graham, Stuart J. H. and David C. Mowery, 2003. "Intellectual Property Protection in the U. S. Software Industry," in Wesley M. Cohen and Stephen A. Merrill, eds., *Patents*

in the Knowledge-Based Economy, National Research Council, Washington: National Academies Press, pp. 219-58.

- Greene, William H. 1993. *Econometric Analysis*, 2nd Ed. New York: Macmillan Publishing Co.
- Griliches, Zvi. 1994. "Productivity, R&D, and the Data Constraint," *American Economic Review*. Vol. 84, pp. 1-23.
- Grindley, Peter C., and Teece, David J. 1997. "Managing Intellectual Capital: Licensing and Cross-Licensing in Semiconductors and Electronics," *California Management Review*, Vol. 39, pp. 8-41.
- Guellec, Dominique and Bruno van Pottelsberghe de la Potterie. 2001. "R&D and Productivity Growth: Panel Data Analysis of 16 OECD Countries," OECD STI Working Paper 2001/3.
- Hall, Bronwyn H. 2003. "Exploring the Patent Explosion," invited lecture at the ZEW Workshop on Empirical Economics of Innovation and Patenting, Mannheim, Germany, March 14/15, 2003.
- Hall, Bronwyn H. and John van Reenen. 1999. "How Effective Are Fiscal Incentives for R&D? A Review of the Evidence," NBER Working Paper No. 7098.
- Hall, Bronwyn H., Adam B. Jaffe, and Manuel Trajtenberg. 2001. "The NBER Patent Citations Data File: Lessons, Insights and Methodological Tools," NBER Working Paper No. 8498.
- Hall, Bronwyn H., and Rosemary Ziedonis. 2001. "The Patent Paradox Revisited: An Empirical Study of Patenting in the U.S. Semiconductor Industry, 1979-1995" *RAND Journal of Economics*. Vol. 32, pp. 101-128.
- Hunt, Robert M. 1999. "Patent Reform: A Mixed Blessing for the U.S. Economy?" Federal Reserve Bank of Philadelphia *Business Review* (November/December), pp. 15-29.

- Hunt, Robert M. 2001. "You Can Patent That? Are Patents on Computer Programs and Business Methods Good for the New Economy?" Federal Reserve Bank of Philadelphia *Business Review*, 1st Quarter, pp. 5-15.
- Hunt, Robert M. "Patentability, Industry Structure, and Innovation," *Journal of Industrial Economics* (forthcoming).
- Jaffe, Adam B. 2000. "The U.S. Patent System in Transition: Policy Innovation and the Innovation Process," *Research Policy*, Vol. 29, pp. 531-557.
- Kortum, Samuel, and Josh Lerner. 1999. "What is Behind the Recent Surge in Patenting?" *Research Policy*, Vol. 28, pp.1-22.
- Levin, Richard C. and Alvin Klevorick, Richard Nelson, and Sidney Winter. 1987. "Appropriating the Returns From Industrial Research and Development." *Brookings Papers on Economic Activity*. Vol. 3: 783-831.
- Machlup, Fritz. 1958. *An Economic Review of the Patent System*. Study No. 15, U.S. Senate Committee on the Judiciary, Subcommittee on Patents, Trademarks and Copyrights. Washington: U.S. Government Printing Office.
- National Research Council. 2003. *Patents in the Knowledge-Based Economy*. Washington: National Academies Press. www.nap.edu
- National Science Foundation. 2003. "Research and Development in Industry: 2000. Arlington, VA: National Science Foundation, Division of Science Resources Statistics.
- USPTO. 1996. "Examination Guidelines for Computer Related Inventions," *Federal Register*, Vol. 61, pp. 7478-92.

Cases

- Diamond v. Diehr* 450 U.S. 175 (1981).
- Gottschalk v. Benson*, 409 U.S. 63 (1972).
- In re Alappat*, 33 F.3d 1526 (1994).

Data Appendix

Identifying Software Patents

We count as a software patent any utility patent (excluding reissues) granted after 1975 which satisfies the following conditions:

1. The terms *software* or *computer* and *program* appear in the specification;
2. The terms *antigen*, *antigenic*, and *chromatography*, do not appear in the specification; and
3. The terms *chip*, *semiconductor*, *bus*, or *circuit*, or *circuitry* do not appear in the title.

Using this algorithm, we identified 130,650 software patents granted in the years 1976 to 1999. For a comparison of this definition, and the resulting patent counts, with others used in the literature, see our working paper.

Matching Patents to Firms

Our statistical analysis relies on the matching of patents to companies in Standard and Poor's *Compustat* database. The majority of our matches are obtained from the NBER Patent Citations Data File. For details on that resource, see the working paper by Hall et al (2001). We supplemented those matches using information graciously provided to us by CHI Research (for information about this data, see <http://www.chiresearch.com/information/customdata/patdata.php3>).

Both of these sources link a numeric assignee number issued by USPTO with an alphanumeric CUSIP code that can be used to identify firms contained in Compustat. In addition we also matched the patents of the 25 largest publicly traded software firms and 100 other large R&D performers not already matched in the data provided by the other sources.

The resulting data set includes patents matched to 4,792 distinct subsidiaries and 2,043 parent firms. Over the 1980-99 period, our sample accounts for 68% of successful U.S. patent applications by domestic non-government organizations (mostly corporations) and 73% of software patents granted to these organizations. The matched firms accounted for 91% of R&D spending reported by U.S. firms in Compustat. These coverage ratios are quite stable over the two decades.

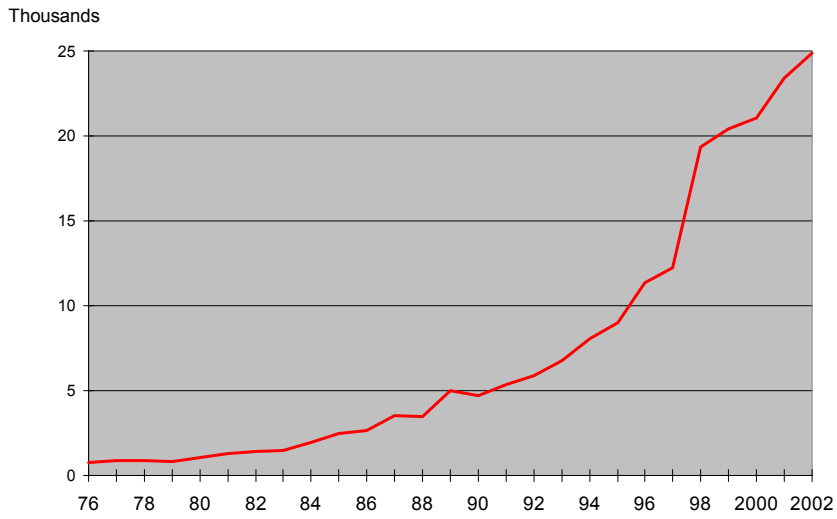
Still, only 37% of R&D performers in the Compustat data set were matched to their patents. This suggests the possibility of *selection bias* in some of our results, because firms successfully matched to their patents may somehow be systematically different from firms who are not matched. In particular, our coverage of the smallest and newest firms in Compustat is not likely to be as good as our coverage of larger and older firms. We conducted a number of statistical tests for selection bias and found this possibility had little or no effect on the results reported here. For details, see our working paper.

Table 1: The Distribution of Software Patents

	(1)	(2)	(3)	(4)	(5)	(6)
	Share of all				All Patents/ R&D	Software patent propensity
	Software patents	Programmers	Programmers & Engineers	All patents		
<u>Manufacturing</u>	75%	11%	32%	88%	3.8	
Chemicals (SIC 28)	5%	1%	2%	15%	2.5	1.5
Machinery (SIC 35)	24%	3%	7%	17%	4.2	4.4
Electronics (SIC 36)	28%	2%	7%	27%	6.8	9.6
Instruments (SIC 38)	9%	1%	4%	11%	7.1	8.7
Other manufacturing	9%	5%	13%	18%	2.3	1.9
<u>Non-manufacturing</u>	25%	89%	68%	12%	3.0	
Software publishers (SIC 7372)	5%	} 33%	18%	1%	1.0	1.0 (all SIC 73)
Other software services (exc. IBM)	2%			1%	2.8	
Other non-manufacturing	4%	55%	49%	4%	3.4	3.8
Addendum: IBM	6%	--	--	2%	5.0	

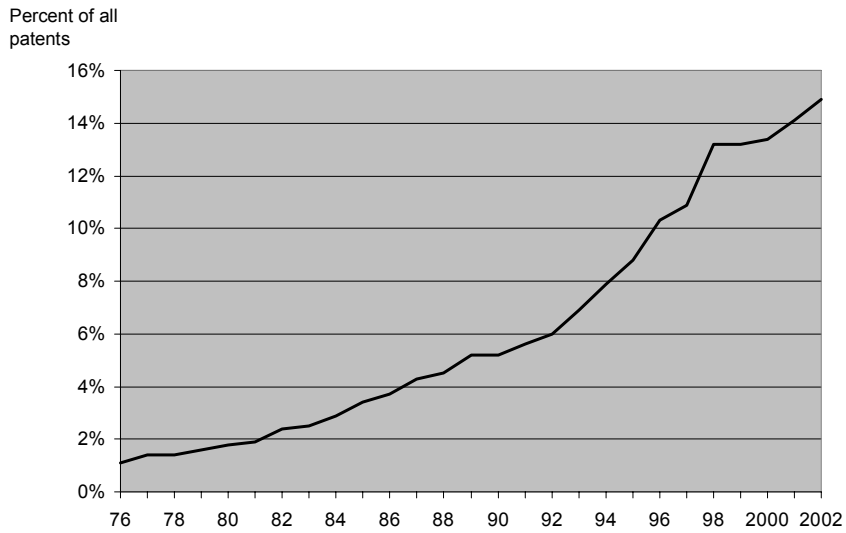
Notes: This table is based on successful patent applications made during years 1994-97 that are matched to a firm in the Compustat data set. Data on computer programmers is from the Bureau of Labor Statistics *Occupational Employment Survey* (various years) and the numbers include system analysts. Note that the tabulation of programmers and engineers for the software sector (SIC 737) includes employment by IBM. The fifth column reports patents granted per \$10 million of R&D in 1996 dollars. The last column reports the relative patent propensity (for software patents) estimated in the statistical analysis contained in our working paper. The numbers are presented relative to the estimated patent of firms contained in the business services sector (SIC 73), excluding IBM. For example, the estimated software patent propensity for the chemical industry is 1.5 times that for SIC 73.

Software Patents Granted



Note: The table plots software patents by grant date, not application date.

Software Patent Share



Note: The table plots software patents by grant date, not application date.