The Software Patent Experiment*

BY ROBERT HUNT AND JAMES BESSEN

Over the past two decades, the scope of technologies that can be patented has been expanded to include many items previously thought unsuitable for patenting, for example, computer software. Today, the U.S. Patent and Trademark Office grants 20,000 or more software patents a year. Conventional wisdom holds that extending patent protection to computer programs will stimulate research and development and, thus, increase the rate of innovation. In this article, Bob Hunt and Jim Bessen investigate whether this has, in fact, happened. They describe the spectacular growth in software patenting, who obtains patents, and the relationship between a sharp focus on software patenting and firms' investment in R&D.

When it comes to patents, the U.S. has undergone an almost accidental process of legal innovation over the last two decades. Standards have been eased: 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 has been increased to include, among other things, gene sequences, computer programs, and methods of doing business.[1]

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 debating 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 research and development (R&D) and, therefore, the rate of innovation.[2] 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.[3] But these surveys also show

Bob Hunt is a senior economist in the Research Department of the Philadelphia Fed.

Jim Bessen is a lecturer at Boston University School of Law.

* An earlier version of this article was published in: *Patents, Innovation, and Economic Performance*, Proceedings of the Conference of the Organization for Economic Cooperation and Development on "IPR, Innovation, and Economic Performance," Paris, August 2003, OECD, Paris (forthcoming).

1 For an examination of the economic effects of changing patent standards, see Bob Hunt's 1999 *Business Review* article. For a brief history of intellectual property rights for computer programs, see Hunt's 2001 *Business Review* article.

2 Many studies examine the relationship between growth in R&D and growth in productivity or economic output. See, for example, the working paper by Dominique Guellec and Bruno van Pottelsberghe de la Potterie and the review article by Zvi Griliches.

3 See, for example, the article by Richard Levin, Alvin Klevorick, Richard Nelson, and Sidney Winter and the working paper by Wesley Cohen, Richard Nelson, and John Walsh. Using data compiled for that working paper, Ashish Arora, Marco Ceccagnoli, and Wesley Cohen present evidence that firms that rate patents as both important and effective tend to do more R&D.

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.[4]

The research described in this article suggests there is some reason for concern about the economic effects of software patents.[5] We found that software patents are not closely related to the creation of computer programs — the vast majority of software patents are obtained by firms outside the software industry.[6] We also found that firms that focus on software patents, in the sense that a higher share of their new patents is 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 R&D is more complicated than is often assumed in the policy debate. In short, we did not find 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 of a century. Some of these changes include 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 more than 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.[7] If the patent is infringed, the patent owner may sue the infringer to recover lost profits. If the infringement was willful, the court may award additional damages.

The American patent system has changed in a number of important ways over the last quarter of a century.

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 the 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 nonobvious. 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 provide instructions on how the nonobviousness requirement should 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.[8] 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.

[4] For recent reviews, see the reports by the Federal Trade Commission, the article by Stephen Merrill et al., and the article by Nancy Gallini.

[5] This article is based on Jim Bessen and Bob Hunt's 2004 working paper.

[6] 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, that is, those classified in SIC 737 (computer programming, data processing, and other computer-related services).

[7] 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.

[8] 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 effect on subsequent litigation, see the article by Adam Jaffe. The economic effects of reduced patentability standards are examined in Hunt's forthcoming article.

Subject Matter Exceptions. As a general principle, the American patent system is not designed to treat different kinds of inventions differently. For example, when 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."[9] These words are often mentioned in judicial decisions where a federal court is confronted with the problem of interpreting Congress's 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, therefore, unpatentable subject matter. But it did not take very long before new decisions began to blur this seemingly bright 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 nonobvious aspects of the invention did not consist entirely of the software. Even this distinction gradually eroded.[10]

Any real difference in the treatment of software and other inventions was eliminated after a 1994 appeals court decision (*in re Alappat*) upheld the patentability of a computer program that smooths 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 its examination guidelines for computer-related inventions. Thereafter, the number of software patents granted increased dramatically (Figure 1).

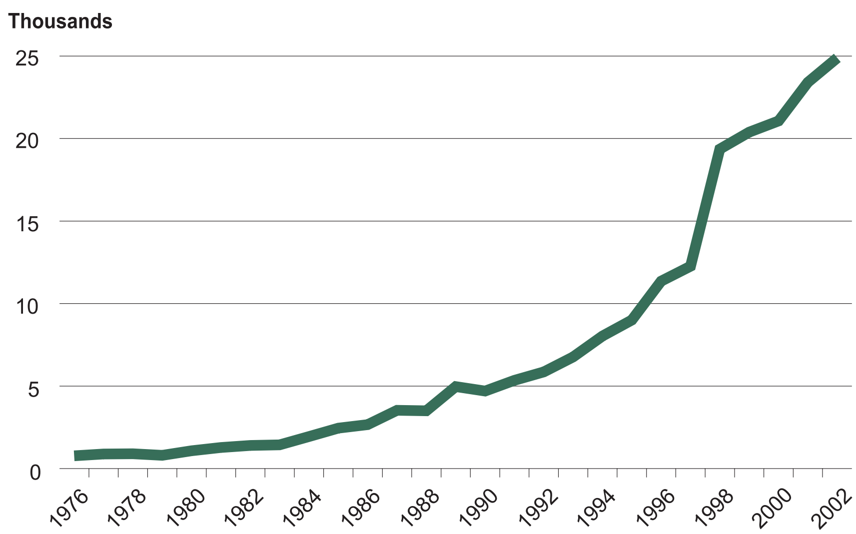
TAKING A CLOSER LOOK AT SOFTWARE PATENTS

Despite considerable interest in the effects of granting patents on computer programs, there is no official list of software patents. The 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 must devise their own ways of identifying software patents.[11] The data used in this article are based on a simple keyword search of the 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.[12]

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 had doubled again 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 counted as software patents increased from about 2 percent in the early 1980s to nearly 15 percent by 2002 (Figure 2).

FIGURE 1

Software Patents Granted in the U.S. (1976-2002)



Source: U.S. Patent and Trademark Office and authors' calculations.
Plots software patents by grant date.

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

[10] For additional information on the changing treatment of software in patent law, see Hunt's 2001 article in the *Business Review.*

[11] For details on the different approaches, see the articles by John Allison and Mark Lemley, and by John Allison and Emerson Tiller, and the one by Stuart Graham and David Mowery.

[12] 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.

Software Patents Are an American Phenomenon. We can learn something about inventors and the owners of patents by examining information contained in the patent document itself. This information reveals that software patents are a (relatively) home-grown phenomenon. During the 1990s, 70 percent of software patents were obtained by inventors living in the U.S.; that is significantly higher than the share of domestic inventors for all other patents (53 percent). Similarly, 70 percent of all software patents owned by companies went to firms headquartered in the U.S.; 51 percent of all other patents owned by companies went to American firms.[13]

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 percent vs. 80 percent). To put it another way, individuals were relatively less likely to obtain their own software patent 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. We obtained detailed financial data on several thousand U.S. firms from Standard and Poor's *Compustat* database, and using some existing databases and our own research, we matched patents to those firms.[14] We then used this information to compare the median firm ranked in terms of (1) the number of software patents obtained and (2) the number of other patents obtained.[15] 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 percent larger ($13 million vs. $9 million). Measured in terms of spending on research and development, it is 68 percent 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 percent of all patents granted to firms included in the *Compustat* file and at most 7 percent of all software patents (Table).[16] Manufacturers accounted for three out of four software patents. The top five firms in terms of software patents obtained in 1995 were 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.

[13] The USPTO data indicate 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 a firm or a government agency.

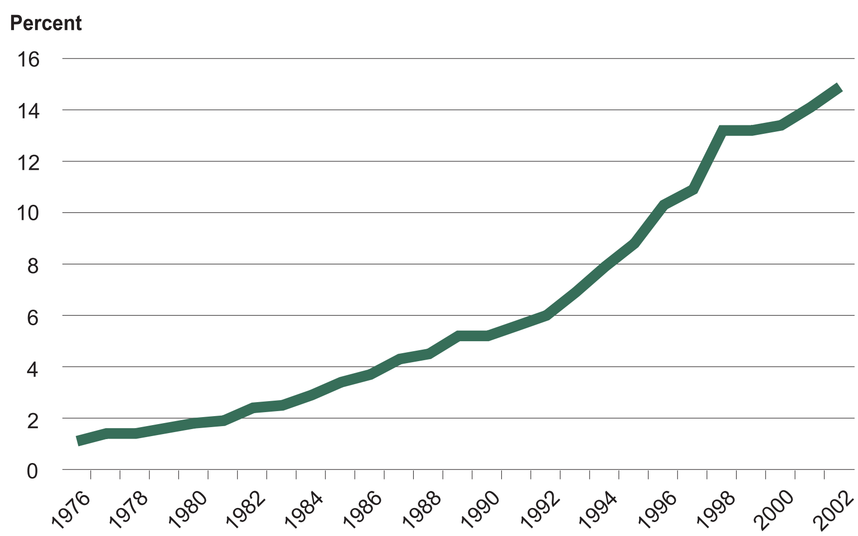
[14] *Compustat* includes information on virtually all firms that file 10-K and 10-Q reports with the U.S. Securities and Exchange Commission. Our matching of patents to firms is based primarily on information contained in the NBER Patent Citations Data File and data generously provided to us by Tony Breitzman of CHI Research. For details on the matching process, see the Data Appendix.

[15] The median identifies the firm where 50 percent of all firms have more patents than it does and 50 percent of all firms have fewer patents than it does.

[16] These statistics are for successful patents applied for during 1994-97. For this calculation, the software industry is defined as firms included in the SIC 737, but excluding IBM, which alone accounted for 6 percent of software patents granted. We treat IBM separately to rule out the possibility that these patterns are the result of a single, large company's activity.

FIGURE 2

Software Patent Share in the U.S. (1976-2002)



Source: U.S. Patent and Trademark Office and authors' calculations.
Plots the percentage of patents granted in a year that are software patents.

TABLE

The Distribution of Software Patents (1994-97)

	(1) Share of all software patents	(2) Share of all programmers	(3) Share of all programmers & engineers	(4) Share of all patents	(5) All patents/ R&D	(6) Software patent propensity
Manufacturing	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
Nonmanufacturing	25%	89%	68%	12%	3.0	
Software publishers (SIC 7372)	5%	33%	18%	1%	1.0	1.0
Other software*	2%#			1%#	2.8#	
Other nonmanufacturing	4%	55%	49%	4%	3.4	3.8
Addendum: IBM	6%	—	—	2%	5.0	

Notes: This table is based on patents issued by the U.S. Patent and Trademark Office applied for during the years 1994-97 and matched to a firm in the *Compustat* data set. Data on computer programmers are from the Bureau of Labor Statistics Occupational Employment Survey (various years) and the numbers include system analysts. 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 Bessen and Hunt's 2004 working paper and described on page 27. The numbers in column 6 are presented relative to the estimated software patent propensity of firms contained in the business services sector (SIC 73). For example, the estimated software patent propensity for the chemical industry is 1.5 times that for SIC 73.

*Firms in SIC 73, excluding those firms in SIC 7372.

#Excludes IBM.

Firms in just three manufacturing industries (machinery, electronics, and instruments) alone accounted for 66 percent of software patents granted to firms — a number that significantly exceeds their impressive 54 percent share of patents of any kind. These numbers are even more remarkable when we examine the distribution of computer programmers across these industries.[17] These are presumably the workers responsible for creating most new computer programs. Manufacturers of machinery, electronics, and instruments employed only 6 percent of all computer programmers and yet they obtained 66 percent of software patents. Firms outside the manufacturing sector employed 90 percent of computer programmers, but together they accounted for only 25 percent of software patents. It would appear that the distribution of software patents across industries reflects something other than the creation of software.

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

WHY ARE THERE SO MANY SOFTWARE PATENTS?

The previous section shows that firms obtain many software patents today, but they either could not, or did not wish to, obtain them in the past. Of course, the software sector of the economy has also grown rapidly over time. But these explanations tell us very little about why firms obtain software patents, and they potentially exaggerate the effects of legal changes by ignoring economic and other factors that may have contributed to the explosion in software patenting. Let's look at the differences in the software-patenting behavior of firms across industries and over time, and let's look for any relationships between a firm's software patenting behavior and its R&D investments.

Estimating the Propensity to Patent. 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 four and seven patents (of any kind) for every $10 million in R&D they spent (see column 5 of the Table). That compares with only about one 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 four to seven more patents by manufacturers of machinery, electronics, and instruments and one additional patent by software companies.

A more sophisticated analysis shows that firms apply for more software patents when they are both more research-oriented and more capital-intensive and when the industry workforce consists of more programmers and engineers.[18] We did not find a difference in the propensity to patent software between old and young firms except in the software industry. There, new firms have a significantly lower propensity to patent software than older firms in the same industry.[19]

Manufacturers, in general, have a much higher propensity to patent software than do firms in the software industry (see column 6 of the Table). After we account for R&D and other factors, firms in the machinery, electronics, and instruments industries obtain software patents at a rate four to 10 times higher than firms in the software sector. In addition, the propensity to patent software is significantly higher for firms in industries in which their peers obtain more patents (of all kinds) per employee. In short, the pattern of software patenting across U.S. firms seems to be more closely related to industry-wide variations in the utilization of patents in general than to the resources devoted to creating software.

Manufacturers, in general, have a much higher propensity to patent software than do firms in the software industry.

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

It is likely that a good part of this increase in the propensity to patent is the result of changes in the legal treatment of software patents.[20] 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 obtains additional patents, the profits

---

[18] This section is based on a regression analysis where we controlled for firm and industry characteristics and allowed for the possibility that patent propensity has changed over time. Details may be found in our 2004 working paper. For an excellent example of this methodology applied to the American semiconductor industry, see the article by Bronwyn Hall and Rosemarie Ziedonis

[19] A new firm in our analysis represents the first five years that a company is reported in *Compustat*.

[20] We cannot attribute all of this residual increase to legal changes because the pattern can also be explained by productivity growth, that is, increases in the number of inventions per programmer. Our review of the available studies suggests that any reasonable estimate of productivity growth in software would explain less than half of the residual increase in the propensity to patent software over time.

earned on its inventions should rise. This should encourage the firm to engage in more R&D. Similarly, when a firm engages in 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 (for example, R&D and software patents) 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. 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.[21] 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 the 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 increase in the number of such patents (the quantity demanded rose as the cost of software patents fell), even after we've controlled for other factors. But what has the effect been on demand for R&D?

The Relationship Between R&D and Software Patents Has Changed. To answer this question, we can 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 focus on *software patents* — its new software patents divided by all its new patents — over time.[22] An increasing focus on software patents 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 focus on software patents would suggest that software patents and R&D are indeed complementary inputs.[23]

Our research shows that, all else equal, during the late 1980s, firms that increased their focus on software patents tended to 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. For the 1990s, we found a much stronger relationship, but it was negative: All else equal, increases in share of software patents were associated with decreases in research intensity. This suggests that in the 1990s, software patents substituted for R&D.[24]

This effect is concentrated in the machinery (including computers) and electronics (including semiconductors) industries and the software industry broadly defined[25] — in other words, among the industries that account for

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.

---

[21] Note that the total value of IT equipment purchased might rise or fall depending on how much more IT equipment is purchased in response to the drop in price. A firm's demand for a good is said to be elastic when a decline in price, expressed as a percent change, induces a larger increase, again expressed as a percent change, in the quantity demanded.

[22] In our statistical analysis, we examined the relationship between changes in firms' R&D intensity and changes in their focus on software patents over five-year intervals. We controlled for changes in input prices (including information technology), size of the firm, new vs. established firms, employment of computer programmers, and idiosyncratic factors specific to the firm or industry.

[23] A positive correlation means that increases in focus on software patents are associated with increases in R&D intensity and decreases in focus on software patents are associated with decreases in R&D intensity.

[24] We also found evidence that this negative correlation had become even more negative by the late 1990s, but we cannot be certain this is true.

[25] 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 focus on software patents and changes in R&D intensity among the remaining firms in SIC 73.

about two-thirds of all software patents. Outside of those industries, there was no systematic relationship, during the 1990s, between an increase in focus on software patents 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 percent 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 five 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 increased focus on software patents 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 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. This conventional wisdom is often cited in arguments that favor extending patent protection to computer programs in Europe. 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 propensities to patent than anything peculiar to software in itself. In general, industries known for prodigious patenting account for the vast majority of software patents obtained. Firms located in such industries have a higher propensity to patent software.

> The increases in the total number of software patents and in share of software patents are consistent with firms' responding to a decline in the relative cost of obtaining these patents or, alternatively, an increase in their cost effectiveness.

The increases in the total number of software patents and in the share of software patents are consistent with firms' responding to a decline in the relative cost of obtaining these patents or, alternatively, an increase in the their cost effectiveness. But the negative correlation between increases in firms' focus on software patents and their R&D intensity in the 1990s suggests that firms may be substituting for R&D with software patents.

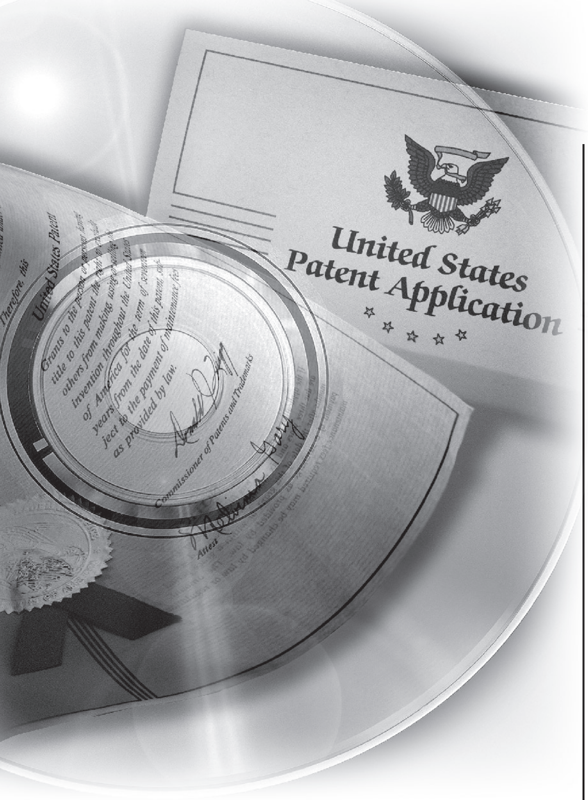
A Productivity Shock. Another hypothesis is that the U.S. economy has experienced a large productivity shock that favored inventions implemented via computer programs. Such a shock would be consistent with a large increase in software patenting and the long-run trend toward higher research intensity among American firms. But it is inconsistent with the negative correlation between increases in share of software patents and R&D intensity. What's more, the observed increase in the propensity to patent software seems too large to be explained 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 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 reduces the cost of doing R&D, and this might explain the observed negative correlation between focus on software patents 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

---

[26] See, for example, the article by Philip Berger and the article by Bronwyn Hall and John van Reenan. In addition, our analysis takes into account changes in the cost of software and the share of computer programmers in the industry workforce.



of R&D that firms engage in increases by at least as much, in percentage terms, as the decrease in its cost. Thus, even if software reduces the cost of research and development, R&D intensity should not fall and might even increase.

Patent Thickets. In contrast to the incentive theory already described, suppose instead that firms in an industry 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 — for example, by demanding royalties — and, in the process, 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 for firms to build larger patent portfolios, especially when their rivals focus on patents as a competitive strategy. Economists sometimes describe this type of environment as a *prisoner's dilemma*.[29] All firms would be better off if they did not act in this way, but each firm would be worse off if it did not respond to a surge in patenting by their rivals. Under these circumstances, firms may find themselves competing 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.

The patent thicket explanation is consistent with the observed rise in propensity to patent and the negative relationship between changes in share of software patents and research intensity in certain industries. It might also explain why software patents are more common in industries with high propensities to patent (machinery, electronics, and instruments) rather than in industries that focus primarily on developing software. Also, it is consistent with the observation that the propensity to patent is higher in industries in which 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 patent system for a long time, it would be irresponsible, on the basis of our present knowledge, to recommend abolishing it."

What would Machlup say 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 of a century.

These changes are often justified on the basis of conventional wisdom: Granting more and stronger property rights will necessarily stimulate innovation. Our evidence suggests this assumption may be incorrect in the case of software patents. If, instead, the legal changes create patent thickets, the result might well be less innovation. BR

[27] For evidence of this phenomenon in the electronics and semiconductor industries, see the article by Peter Grindley and David Teece and the one by Bronwyn Hall and Rosemarie Ziedonis.

[28] For a theoretical model of this intuition, see the 2003 working paper by James Bessen.

[29] The term is derived from the example of two suspects arrested and interrogated separately. If they both remain silent, the prosecutor has little evidence, and each will receive a small penalty. If one suspect rats and the other doesn't, the rat will reduce his own punishment, but the silent one will be punished severely. Knowing each other's incentives, both suspects rat on each other.

Data Appendix

Identifying Software Patents

We count as a software patent any utility patent (excluding reissues) granted after 1975 that 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 Bronwyn Hall, Adam Jaffe, and Manuel Trajtenberg). We supplemented those matches using information graciously provided to us by CHI Research (for information about these data, see www.chiresearch.com/information/customdata/patdata.php3.

Both of these sources link a numeric assignee number issued by the 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 period 1980-99, our sample accounts for 68 percent of successful U.S. patent applications by domestic nongovernment organizations (mostly corporations) and 73 percent of software patents granted to these organizations. The matched firms accounted for 91 percent of R&D spending reported by U.S. firms in *Compustat*. These coverage ratios are quite stable over the two decades.

Still, only 37 percent 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 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.

REFERENCES

Allison, John R., and Mark A. Lemley. "Who's Patenting What? An Empirical Exploration of Patent Prosecution," *Vanderbilt Law Review,* 58, 2000, pp. 2099-2148.

Allison, John R., and Emerson H. Tiller. "An Empirical Analysis of Internet Business Method Patents," mimeo, University of Texas at Austin, 2001.

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

Berger, Philip. "Explicit and Implicit Effects of the R&D Tax Credit," *Journal of Accounting Research*, 31, 1993, pp. 131-71.

Bessen, James. "Patent Thickets: Strategic Patenting of Complex Technologies," ROI Working Paper, 2003.

Bessen, James, and Robert Hunt. "An Empirical Look at Software Patents," Federal Reserve Bank of Philadelphia Working Paper 03-17/R, March 2004.

Cohen, Wesley M., Richard R. Nelson, and John P. Walsh. "Protecting Their Intellectual Assets: Appropriability Conditions and Why U.S. Manufacturing Firms Patent (or Not)," NBER Working Paper 7552 (2000).

Federal Trade Commission. *To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy.* Washington, DC: Federal Trade Commission, 2003.

Gallini, Nancy T. "The Economics of Patents: Lessons from Recent U.S. Patent Reform," *Journal of Economic Perspectives*, 16, 2002, pp. 131-54.

Graham, Stuart J.H., and David C. Mowery. "Intellectual Property Protection in the U.S. Software Industry," in Wesley M. Cohen and Stephen A. Merrill, eds., *Patents in the Knowledge-Based Economy,* Washington, D.C.: National Academies Press, 2003, pp. 219-58.

Griliches, Zvi. "Productivity, R&D, and the Data Constraint," *American Economic Review,* 84, 1994, pp. 1-23.

Grindley, Peter C., and Teece, David J. "Managing Intellectual Capital: Licensing and Cross-Licensing in Semiconductors and Electronics," *California Management Review,* 39, 1997, pp. 8-41.

Guellec, Dominique, and Bruno van Pottelsberghe de la Potterie. "R&D and Productivity Growth: Panel Data Analysis of 16 OECD Countries," OECD STI Working Paper 2001/3, 2001

Hall, Bronwyn H., and John van Reenen. "How Effective Are Fiscal Incentives for R&D? A Review of the Evidence," NBER Working Paper 7098, 1999.

Hall, Bronwyn H., Adam B. Jaffe, and Manuel Trajtenberg. "The NBER Patent Citations Data File: Lessons, Insights and Methodological Tools," NBER Working Paper 8498, 2001.

Hall, Bronwyn H., and Rosemary Ziedonis. "The Patent Paradox Revisited: An Empirical Study of Patenting in the U.S. Semiconductor Industry, 1979-1995," *RAND Journal of Economics*, 32, 2001, pp. 101-28.

Hunt, Robert M. "Patent Reform: A Mixed Blessing for the U.S. Economy?" Federal Reserve Bank of Philadelphia *Business Review* (November/December 1999), pp.15-29.

Hunt, Robert M. "You Can Patent That? Are Patents on Computer Programs and Business Methods Good for the New Economy?" Federal Reserve Bank of Philadelphia *Business Review,* First Quarter, 2001, pp. 5-15.

Hunt, Robert M. "Patentability, Industry Structure, and Innovation," *Journal of Industrial Economics* (forthcoming).

Jaffe, Adam B. "The U.S. Patent System in Transition: Policy Innovation and the Innovation Process," *Research Policy*, 29, 2000, pp. 531-57.

Kortum, Samuel, and Josh Lerner. "What Is Behind the Recent Surge in Patenting?" *Research Policy*, 28, 1999, pp. 1-22.

Levin, Richard C., Alvin Klevorick, Richard Nelson, and Sidney Winter. "Appropriating the Returns from Industrial Research and Development," *Brookings Papers on Economic Activity*, 3, 1987, pp. 783-831.

Machlup, Fritz. *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, 1958.

Merrill, Stephen A., Richard C. Levin, and Mark B. Myers, eds., *A Patent System for the 21st Century.* Washington: National Academies Press, 2004, www.nap.edu

National Science Foundation. "Research and Development in Industry: 2000," Arlington, VA: National Science Foundation, Division of Science Resources Statistics, 2003.

USPTO. "Examination Guidelines for Computer-Related Inventions," *Federal Register,* 61, 1996. 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).