Applicability of Patent Protection to Software Inventions

Jason Chalecki
Anoop Gupta
Dave Kong
John Li
Jerry H. Lin

I. Patent Backgrounder

Patents and Copyright

In the United States, the software industry uses the patent system to protect its intellectual property rights. The patent system protects the ideas themselves, and a patent owner can prevent others from making, using, or selling a patented invention. In the domain of software, patent protection extends to algorithms or programming techniques.

Copyrights, on the other hand, only prevent copying the *expression* of an idea. In the domain of software, this means the exact duplication of the software or of any source code that is a part of the final product. Copyright also protects an inventor from duplication of software by simple mimicry of an existing product.

The problem with relying solely on copyright protection is that recently, courts have interpreted copyright law very narrowly. The premise upheld by the courts is that a copyright will protect the expression of an idea, but *not* the idea itself. Patents, on the other hand, preclude others from simulating the idea or implementing it in a slightly different way. For instance, a software patent will make it illegal for a rival company to implement a particular algorithm using a different programming language.

The History of Patents

The current patent system in the United States has evolved greatly as circumstances and industries have grown and changed.

The origin of patenting dates back to medieval times, when the sovereign of a land granted exclusive rights in order to raise money without raising taxes. Some of these grants applied to innovations and to manufacturing. The first modern concept of the patent dates back to 1449, when King Henry VI awarded a patent for stained glass manufacturing. While in fifteenth century England there was nothing novel about the art of stained glass making, the monarchy recognized the value of protecting certain arts and industries, including those that were imported from other parts of Europe.

Around 1552, a series of "letter patents" were issued by the monarchy in England for the benefit of the crown and its friends and associates. These patents protected entire industries and soon enough the power and wealth in these industries shifted to individuals and groups favored by the current monarchy. Reform of these practices began during the reign of Queen Elizabeth II, when the rule became that the queen would grant patents that she deemed useful to the entire nation. Around 1624, in response to abuses of power by the monarchy, Parliament passed the English Statute of Monopolies which "outlawed all royally sanctioned monopolies." Realizing the importance of protecting inventors and the economic benefits associated with encouraging innovation, an exception was made for patents of "new manufactures." These patents were awarded to the inventor as long as their new devices did not hurt trade or result in price increases, and had a limit of fourteen years.

Much of the United States patent system found its basis in the already established English version. The premise behind both systems was that there was an overall benefit to society to develop and encourage new inventions. From the Constitution of the United States:

> Congress shall have the Power... To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries.

In 1790, President Washington stated in his first address to Congress:

> I cannot forbear intimating to you the expediency of giving effectual encouragement, as well to the introduction of new and useful inventions from abroad as to the exertion of skill and genius at home.

Shortly after, Congress passed the first United States Patent Act. This act had seven sections and allowed any two of the Secretary of State, the Secretary of War, and the Attorney General together to grant patents for up to fourteen years for inventions that were useful and important, as long as the inventor provided specifications describing the invention to the Secretary of State. Up to this point, the act did not allow for patents from abroad.

This act was repealed in 1793 and replaced with one drafted primarily by Thomas Jefferson, who was Secretary of State at the time and was therefore involved heavily with patents and the effects of the 1790 act. The new act was more specific in what could and could not be patented—a standard that has remained almost unchanged to the present day:

> …any new and useful art, machine, manufacture or composition of matter and any new and useful improvement on any art, machine, manufacture or composition of matter.

In 1800, the 1793 patent act was amended to allow foreigners who were residents of the United States for longer than two years to apply for patents, and also allowed triple damages for patent infringement. Also, after 1800, the courts ruled that the term "new," as it applies to inventions, simply meant that the invention had to be previously unpatented. In 1829, the Supreme Court changed this in the case of *Pennock v. Dialogue*, when they realized the danger in allowing an inventor to postpone filing a patent until there was any competition, and began to deny patent protection if the invention was already publicly used before applying for protection.

Between 1793 and 1836, there was a large increase in the number of patents being filed. This increase brought to the forefront many of the problems with the patent system, and resulted in the passing of another patent act. This act created a patent office for the first time that was separate from the regular duties of the Secretary of State. In addition, a system to distribute newly issued patents was developed to provide public access to the knowledge in those patents. The number of patents being filed went up as a result, and so did the quality of the patents being filed. Another important problem addressed by the new act was the lifetime of patents. Eli Whitney, for example, had trouble with his patent on the cotton gin because by the time he was able to enforce his patent, the fourteen year life span had already run out. The 1836 act allowed for the commissioner of the patent office to extend the life of a patent an extra seven years.

The next major step forward took place in 1850, when it was ruled that patents not only had to be new and useful, but also non-obvious. This new condition was introduced by the Supreme Court, rather than by legislation, in the case of *Hotchkiss v. Greenwood.*

Under the revised Patent Act (US Code Title 35), exclusive rights were awarded to "whoever invents or discovers any new and useful (1) process, (2) machine, (3) manufacture, or (4) composition of matter, or any new and useful improvement thereof." (**utility**)

Section 102, 103 specified the aforementioned restrictions as follows:

- The invention must not be known or used in this, *or a foreign country*, prior to the invention by the applicant. (**novelty**)

- The invention must not be in public use or on sale in this, *or a foreign country*, twelve months prior to the application for patent. (**statute of limitation**)

- The invention must not be obvious to a person having ordinary skill in the art. (**non-obviousness**)

The courts have further established that laws of nature, scientific phenomena, and mathematical formulae are excluded from patentability, as exclusive rights to such fundamental "scientific truths" of our world would grant unreasonable control to individuals.

It is under this policy that industry learned to take advantage of patents. Some individuals became very wealthy from royalties on their patents, and some companies became very powerful. Some examples are Colt (revolver), Yale (lock), and Eastman (camera).

Since there weren't any major changes in the system from this point on, we now shift focus to patents as they apply to the software industry.

Software Industry Meets Patents

Up through the 1970s, the U.S. patent office avoided granting any patents that involved a calculation made by a computer. At that point, patents had only been granted to processes, machines, manufactured items, and compositions of matter; computer programs were viewed merely as mathematical algorithms. The notion that patents could be granted for scientific fact or mathematical expression of that fact seemed beyond the scope of patents.

In 1972, the United States Supreme Court confronted this question in the case of *Gottschalk v. Benson*. The Court framed the issue this way: "The question is whether the method described and claimed is a 'process' within the meaning of the Patent Act." The Court went on to describe the claimed invention, which was a method for converting the encoding of some numbers:

> The patent sought is on a method of programming a general-purpose digital computer to convert signals from binary-coded decimal form into pure binary form. A procedure for solving a given type of mathematical problem is known as an 'algorithm.' The procedures set forth in the present claims are of that kind; that is to say, they are a generalized formulation for programs to solve mathematical problems of converting one form of numerical representation to another. From the generic formulation, programs may be developed as specific applications.

Even though the Court decided that this invention was not patentable, they did not rule on whether *any* computer program would ever be patentable.

In 1976, the Court of Customs and Patent Appeals ruled on a patent application for dynamically rearranging priorities in a multi-program operating system. The patent office had denied the patent request because they interpreted *Benson* as saying that *no* computer programs could be patented. The Court, on the other hand, believed that there was a difference between a mathematical algorithm and *any other* algorithm.

> Over-concentration on the word 'algorithm' alone, for example, may mislead. The Supreme Court carefully supplied a definition of the particular algorithm before it, i.e., '[a] procedure for solving a given type of mathematical problem.' The broader definition of algorithm is 'a step-by-step procedure for solving a problem or accomplishing some end.' *Webster's New Collegiate Dictionary* (1976). It is axiomatic that inventive minds seek and develop solutions to problems and step-by-step solutions often attain the status of patentable invention. It would be unnecessarily detrimental to our patent system to deny inventors patent protection on the sole ground that their contribution could be broadly termed an 'algorithm.'

The Court decided to grant the patent in this case because the idea being patented was not a mathematical algorithm, but a general algorithm. The Court ruled this way again in 1978 on a patent application for a new method of typesetting alphanumeric information using a computer-based control system in conjunction with a phototypesetter.

A real shakeup occurred in 1980, in the case of *Diamond v. Diehr*. In this case, the Supreme Court ordered the U.S. patent office to grant a patent for an invention that used computer software. The invention in this case was a method for figuring out how to optimally heat rubber for curing. It used a computer to control and time the heating, but the invention also included steps related to the heating itself as well as the removal of the rubber from heat. The Supreme Court contended that the invention was not merely a mathematical algorithm and was therefore patentable, even though the only new part to the process was the computer control of the timing.

The petitioner in this case came up with a general definition of what could be patented:

> The term 'algorithm' is subject to a variety of definitions. The petitioner defines the term to mean:
>
> A fixed step-by-step procedure for accomplishing a given result; usually a simplified procedure for solving a complex problem, also a full statement of a finite number of steps.
>
> A defined process or set of rules that leads and assures development of a desired output from a given input. A sequence of formulas and/or algebraic/logical steps to calculate or determine a given task; processing rule.

By adopting this definition, *Diamond v. Diehr* significantly broadened the definition of what was patentable. Thus, a claim cannot be denied solely because it contained mathematical formulae, instead, the invention as a whole must be considered. Still, a mathematical algorithm itself cannot be patented.

It didn't take long until another major shift in software patentability to occur. In St*ate Street Bank & Trust Company v. Signature Financial Group*, the Court of Appeals for the Federal Circuit effectively threw out the mathematical algorithm exception. It was determined that "mathematical algorithms were patentable as long as their application produced a 'useful, concrete, and tangible result'".

From 1981 to the early 1990s, the U.S. patent office was left to figure out what was merely a mathematical algorithm and what was a patentable invention that involved a mathematical algorithm—all with the help of the inventors themselves. Some of the lower courts tried to clarify what was patentable, but in the end they only confused the process even more. After years of this, it became clear that the patentability of an invention using some form of software was not being decided by the inventor or the patent office, but rather by the patent attorney who filed the patent.

In the early 1990s, the Federal Circuit Court tried to clear up this confusion. Their premise was that the invention should be examined *as a whole* when deciding on its patentability. If the entire process was simply a mathematical algorithm, then the invention was not patentable; if, on the other hand, the invention used software to manipulate and compute numbers that represented real world values, then the invention was patentable because it related to some real world process or concept.

Finally, in 1995, the patent office decided to weigh in on the subject and set down some guidelines for their courts. They came up with the *Final Computer Related Examination Guidelines* in 1996, which applied to both hardware and software related patents. Their premise was that certain kinds of software inventions are patentable:

> Those having "significant post solution activity," meaning that the software program is used to control something external to the software routine (such as curing rubber).
>
> Those having "pre-computer process activity," meaning software programs that manipulate numbers representing concrete, real world values (such as electrocardiograph signals and seismic measurements.
>
> Those claimed in connection with a specific machine or product. This can be accomplished by defining specific code segments or routines in the patent application, or by claiming the invention in connection with a specific type of computer or memory structure.

Additionally, software patents still needed to meet the basic, general requirements: they must be statutory, new, useful, and non-obvious. These new guidelines, nevertheless, made most software inventions patentable.

Of the requirements, obviousness is probably the most difficult to assess. This problem and the newly posed concerns that allowing software patents might actually harm society are the two biggest challenges facing our thinking about patents today.

II. Comparative Analysis: Patent Laws Around The World

In this section, we will compare and contrast how other countries and regions approach the issue of patents. Our focus will be on the European Union and Japan. Together with the United States, they form the three largest economic entities in the world. Furthermore, they also have the most sophisticated and complete framework of laws concerning patents, copyrights, and intellectual property rights in general.

We will also take a look at China, which has had the world's fastest growing economy for the past decade. China's plunge into the global economy demands considerable revision to its laws in order to bring its business environment closer to the established global practices, especially in those areas surrounding intellectual properties.

We begin with a study of fundamental patent laws and processes in these countries/regions, followed by a closer examination of their treatment of software and business method patents. Finally, a comparison chart of the four patent systems is included to indicate key contrasts.

Basic Patent Laws and Processes

United States

As already presented in Part I of this paper, the United States patent system requires (1) utility, (2) novelty, and (3) non-obviousness in order for a patent to be issued. Additionally, its unique first-to-invent system gives the inventor a twelve month grace period to file for patent, enforcing a statute of limitation.

European Union

The European Union enacted the European Patent Convention as a unifying framework of patent laws for its member states. Section 52 defines patentable inventions as "any inventions susceptible of industrial application, which are new and which involve an inventive step."

The Convention does not define categories of patentable inventions as the U.S. counterpart does. The requirements are simply: (a) industrial applicability; (b) novelty; and (c) inventive step. This basically mirrors the U.S. patent requirements of utility, novelty, and non-obviousness.

The EU's definition of a patentable invention is somewhat misleading. The words "industrial application" can carry a broad implication. Many would argue that software clearly has industrial application, so it must be patentable. However, that is not true. Instead, an invention must also be of "technical nature" in order to be patentable.

Article 52, paragraph 2 of the Convention specifically lists what are NOT patentable:

- discoveries, scientific theories and mathematical methods.
- aesthetic creations;

- schemes, rules and methods for performing mental acts, playing games or doing business, and programs for computers;

- presentations of information.

Another major difference is that the EU employs a first-to-file test when determining priority, as opposed to first-to-invent in the U.S. Thus, each invention must not be known or in use prior to its patent filing. If an invention has already been filed in a treaty country prior to filing its claim at the EPO, then the filing date in that country will be used as the priority date.

It must be pointed out that although the European Patent Office issues patents that are honored by all member states, each country still maintains its own patent office and applies its own laws. In most cases, individual countries simply re-issue the same patent after the EPO has examined and issued the original one.

Japan

The first patent law was passed in Japan in 1959—Law No. 121. Section 29 gives the patent office the power to award a patent to "any person who has made an invention which is industrially applicable." "Invention" in this Law means "the highly advanced creation of technical ideas by which a law of nature is utilized," except:

- When inventions were publicly known or worked in Japan prior to filing

- Where an invention could easily have been made … by a person with ordinary skill in the art

In essence, Japan's basic requirements for patentable inventions are almost identical to those of the U.S. and EU. What is quite different is the patent application process.

The inventor seeking patent first files an application with the Japanese Patent Office. Unlike its Western counterparts, the JPO does not automatically examine those applications. Instead, the unexamined applications are published 18 months after their earliest priority date. Within seven years, the applicant may request an examination. The rationale for this is that applicants may use this period to observe market conditions. If the invention turns out to have little market value, then an examination will not be necessary. Prior to issuing the patent, the invention is published for a three month opposition period, during which any party may challenge the patent.

In 1999, Japan passed an amendment that brought its patent law into more agreement with U.S. and EU laws. Three major changes were:

- The term limit for requesting examination was shortened from the original seven years to three years.

- The bar was raised for the novelty requirement, stating that an invention must not be publicly known or used in Japan, *or a foreign country*, prior to filing.

- The three month pre-grant opposition period was changed to a six month post-grant opposition. That is, after examination, the invention is published and open to challenges for six months, but the patent is granted and remains valid until any challenge succeeds.

China

The Patent Law of the People's Republic of China was introduced rather recently (1985). (As a side note, the PRC Trademark Law came into effect in 1983, and the first Copyright Law was enacted as late as 1991.) Even though China was late to the game, its government has steadily amended those laws to bring them inline with international standards, such as GATT Agreement on Trade-related Aspects of Intellectual Property Rights.

Given this extremely short history of legal tradition concerning intellectual property rights, it is not surprising that there still exists much ambiguity and inconsistency in China's patent/copyright laws. The lack of a sufficient body of legal precedents in this area also makes it difficult to fully understand the implications of Chinese laws. It is worth pointing out that Chinese courts differ from Western courts in that they rarely rely on case laws (common law), but instead base their decisions on strict interpretation of statues and codes (civil law).

Similar to the EU and Japan, China employs a first-to-file system and also requires absolute novelty as a prerequisite (no prior use or knowledge in any country). Its usefulness and non-obviousness requirements are also very similar to the other countries.

In 1992, China drastically changed its patent law and expanded the scope of patent protection to include "all types of technological inventions, whether new products or new techniques, including pharmaceutical products and substances obtained by means of a chemical process, foods, beverages and flavorings." Prior to this, China did not protect chemical or pharmaceutical inventions, believing that these were more akin to discoveries rather than inventions.

In addition to expanding the scope of protection, the length of protection is also extended to 20 years from 15, in order to further conform to the global standards. As a final touch, China further loosened the compulsory patent licensing clause so that there is less risk of patent holders being forced to share out their technologies.

Treatment of Software Patents and Business Methods

United States

While initially viewing software as mathematical formulae (and hence not patentable under U.S. law), cases such as *Gottschalk v. Benson*, *Diamond v. Diehr*, and *State Street Bank & Trust Company v. Signature Financial Group*, gradually made software patentable in the United States.

It appears that, by far, the U.S. has the fewest restrictions for what is patentable. The courts no longer emphasize that only tangible things or inventions which have real, physical effects may enjoy patent protection. Software, while naturally being protected by copyright laws, can also be afforded even stronger protection under patent laws. This looseness is further extended to business methods. In reality, the two are often closely linked: in today's technology driven environment, business processes are usually implemented by software and often rely on the Internet, as in the case of e-commerce.

European Union

As indicated in Article 52, paragraph 2 of the Convention, "mathematical methods," "presentation of information" and particularly "programs for computers" as well as "methods for…doing business" are clearly detrimental to the patentability of software. The courts have also rejected computer-related inventions because they supposedly only automated "mental tasks".

The court later reversed, noting that digital image processing is not an abstract process but a "real world activity", stating that "even if the idea underlying an invention may be considered to reside in a mathematical method, a claim directed to a technical process in which the method is used does not seek protection for the mathematical method as such." The Board rationalized its decision, stating that "[t]here can be little doubt that any processing operation on an electric signal can be described in mathematical terms" and that "there is no basis in the E.P.C. for treating digital filters differently from analogue ones [which are patentable]".

In reality, the EU is approaching the U.S. in terms of patentability of software and business methods. As argued by many observers, this is mostly due to the plethora of U.S. companies filing for such patents in Europe and subsequently challenging court decisions regarding the patentability of these particular subject matters.

Japan

Previously, software could only be protected if it was stored on physical media such as CD-ROM or disks, and be sold as a "tangible" product. It was unclear whether software should be treated as a product or a method (i.e. process).

In 2002, the Japanese Diet changed the patent law, greatly strengthening legal protection for software. Under the new law, software is a tangible product in its own right. Therefore, programs not stored on CD-ROMs or disks can now receive patent protection, such as those stored on a network, or purchased via the Internet, etc.

The impact of this change goes beyond just the treatment of software. As we have noted earlier, business methods and software are often closely intertwined in today's business environments.

The Japanese Patent Office has traditionally held the belief that inventions must involve "technological thought". An examination guideline issued in 1999 stated that patents should not be issued for pure business methods, but only for those that had a "technology aspect". Further guidelines issued in the following year emphasized the importance of an inventive step. If the business method invention merely combines means and processes already commonly known to those in the business field, it will not be granted patent protection.

A great side effect of the 2002 revision is that business methods that are implemented by computer software are now patentable. This effectively reversed Japan's patent policy on business methods.

It seems that Japan has always closely followed the U.S. in many respects regarding patent policies. However, most agree that the Japanese Patent Office has a somewhat higher standard for the "inventive step" or "non-obviousness" requirement than its U.S. counterpart.

China

Similar in respect to the U.S., China chose to protect software under copyright law rather than patent or contract laws. Perhaps unique to China, the "Computer Software Protection Rules" was enacted in 1991 to specifically address the issue of software protection.

As mentioned before, there exists much ambiguity in the Chinese laws. One such ambiguity stems from the "knowingly" standard in Article 32 of the Computer Software Protection Rules.

> Where the holder of software does not know or has no reasonable grounds for knowing that the software is infringing (upon an existing copyright), liability for infringement shall be borne by the supplier of the infringing software.

This greatly deviates from the other IP laws. Ignorance of infringement could be a valid defense in Chinese courts.

While software is protected under Chinese copyright laws, it is NOT protected under the patent laws. In fact, software, business methods, methods of diagnosing or treating diseases, and many plant varieties remain unpatentable in China.

Comparison chart

	United States	Europe	Japan	China
Requirements				
Who can file	First to invent	First to file	First to file	First to file
Usefulness	Yes	"Industrial applicability"	Yes	Yes
Novelty	1 yr grace	Absolute novelty	Absolute novelty	Absolute novelty
Non-obviousness	Yes	Yes	Yes	Yes
Technical aspect	No	Yes	Yes	Yes
Duration				
Inventions patents	20 yrs	20 yrs	20 yrs	20 yrs
Design patents	14 yrs	25 yrs	15 yrs	10 yrs
Utility model patents	N/A	10 yrs (currently being unified)	6 yrs	10 yrs
Patentability				
Machines, apparatus	Yes	Yes	Yes	Yes
Chemicals, Pharmaceuticals	Yes	Yes	Yes	Yes (since 1991)
DNA, RNA	Yes	Yes, but can't patent "discoveries"	Yes, has restriction concerning public health issues.	No
Surgical methods	Yes	No	No	No
Software	Yes	Yes	Yes	No
Business methods	Yes	Yes technically, but most restrictive	Yes, bar is between the U.S. and the EU	No
Mathematical algorithms	Not by itself.	No	No	No
Laws of nature, pure scientific truth, math formulae	No	No	No	No

Summary of Key Differences

Most countries seem to be converging on a common set of standards regarding intellectual property laws, especially in the area of patent laws, which are often the most controversial. While the top three players, the U.S., the EU, and Japan, have converged considerably through amendments in the last decade or so, China, representative of newcomers to the global market, is still noticeably behind but catching up quickly.

Some of the major differences between these four patent systems are:

The U.S. has a first-to-invent system with a one year statute of limitation, while everyone else typically use a first-to-file system and require *absolute novelty*.

- Japan has a very different application process for patents.
- The U.S. is, by far, the least restrictive when considering patents for software and business methods. Japan follows in second, while the EU is the most restrictive.
- Software and business methods are not patentable in China.
- The litigation process is significantly different in China due to its lack of emphasis on case laws and precedents.

III. An Analysis

As much as it is important that we know where we have been, it does not tell us where we should go. We now turn to a selection of academic and industrial research that will hopefully make that question less enigmatic through economic analysis backed, in some cases, with empirical studies.

Unconventional Wisdom

Traditional intellectual property protection schemes equate imitation with copying. In a paper by James Bessen and Eric Maskin, it is suggested that "both individual publishers and society more generally may benefit from weak intellectual property enforcement and protection" in an interactive and dynamic environment, such as the Internet. This is because, when innovations are sequential, imitation means an addition of not insignificant value. Additionally, there is little or no evidence to suggest that imitation inhibits innovation in a "highly dynamic environment with rapid sequential innovation".

James Bessen and Eric Maskin state that improved intellectual property protection schemes would need several changes. It would need to recognize the distinction between "creative imitation" and copying. It would probably need to behave differently in dynamic, sequential environments than in static environments. Finally, it would need to recognize that the contribution of subsequent inventors is unpredictable and private information of that inventor. Consequently, the authors conclude that an ideal scheme would be weak enough to encourage much cross-licensing while still preventing direct copying.

When Competition Helps

When innovations in an industry are typically sequential, i.e. an innovation is predicated on past innovations, and complementary, i.e. each additional inventor increases the probability of a successful innovation, James Bessen and Eric Maskin found that "a firm's profit may actually be enhanced by competition, and a patent system may interfere with such competition and with innovation". It may be beneficial for society for there to be imitation of inventions under the conditions that innovation is both sequential and complementary because it will generally help the inventor to discover new innovations. Additionally, this second inventor may have information that is unavailable to the original inventor, thus increasing the overall rate of innovation.

Defenders of the patent system do have a counterargument that, on its face, sounds convincing. If a patent is locking up information that would enable further innovation to occur, the patent holder typically has incentives to license the patent to the other innovator(s). This would allow the innovation to continue. This counterargument neglects to consider the likely scenario where determination of the appropriate licensing fee requires information that is private to the licensee, such as the costs of applying the license or the potential revenue that can be earned. This results in a not insignificant probability that the patent holder will set a license fee that is too high, preventing licensing and, consequently, further innovation to occur.

It seems counterintuitive that imitation can be in a firm's best interests because it will almost always reduce the current profits of the firm. However, imitation will increase the chances of further innovation in the industry, which generally improves the firm's future profits. This can happen one of two ways. If there is no patent protection, the firm can now imitate these later innovations. If there is patent protection, the firm will be able to garner more licensing fees due to the increased number of subsequent innovations that built on its initial innovation.

Bessen and Maskin modeled the situations where there was a sequence of innovations, each predicated on the previous innovation, and where there was a single isolated innovation. They found that there was a larger incentive to a firm to innovate in the model where innovation was sequential than in the single innovation model. The second firm in the model had more incentive to conduct research and development because those efforts increased their chances of making the next innovation, but more importantly it increased their chances of making each subsequent innovation. The complication that patents add in the model with sequential innovation is that a patent holder can hold up subsequent innovation. Licensing of the patent will neutralize this, but the patent holder may not have the information necessary to set an optimal licensing fee. They conclude that, while a firm engaging in research and development activities when there is a single innovation are worse off under imitation, the converse is true with sequential innovation, provided the imitators will conduct research and development.

There is empirical evidence indicating that the model with sequential innovation applies more readily to the high-tech industry. Michael Gort and Steven Klepper gathered data on the innovations that occurred for twenty-three major products. They also collected information on firm entry for those lifetimes. They found that, on average, a product would experience nineteen subsequent innovations. They hypothesized that the rate of innovation would be correlated to the rate of firm entry because when new firms enter a market, they can introduce new technologies. They found their hypothesis to be heavily supported as the highest rates of innovation occurred during and immediately after the period of greatest firm entry into the markets for the products.

There was a natural experiment to test how patents affect behavior under sequential innovation with the drastic change in patent policy during the 1980s. The end of the 1980s saw a leveling off, if not a reversal, of the trend of increasing research and development activity for the previous decade. James Bessen and Eric Maskin did not find even a ten percent increase in research investment for firms acquiring software patents. It is possible that this is the result of a simultaneous decrease in technological opportunities combined with an increase in research and development costs. Bessen and Maskin felt this possibility was refuted by available evidence. They examined whether price effects might have caused research and development intensity. They did not find any evidence that that was the case. They concluded that "the extension of patent protection to software did not generate a relative increase" in research and development activity as predicted by traditional patent theory. To summarize all of their findings, patents "foster innovation incentives" when innovations are isolated, but with sequential innovations, there is incentive enough without patents. Moreover, patents may actually reduce sequential innovations.

A Retrospective Of Recent Patent Policy

There is evidence that, in general, industry has not achieved the optimal level of investment in research and development. In most cases, manufacturing the best product requires the use of multiple ideas from multiple parties. Typically, these ideas will be protected with intellectual property rights. Incentives do exist for the parties to reach a license, but the transaction costs of doing so can be prohibitive. (Two historical examples of note involve the airplane and the radio.) Consequently, it may happen that relaxing the standards for acquiring a patent may actually reduce the amount of investment a firm makes in its research and development. This is more likely in the high tech sector.

> A firm invests in research and development based on the expectation of returns. Patent policy will influence this expected return in the following ways. First, it will affect the likelihood that a discovery can be protected by a patent. Additionally, it

> will affect the profits earned over the lifetime of the patent. If the standard of non-obviousness is reduced, thereby making a patent easier to obtain, two contradicting effects will occur. First, more innovations will be able to be protected by patents, increasing their profitability. Conversely, profits will be smaller and shorter-lived because other firms will also acquire more patents reducing the firm's ability to imitate its competitors. Research by Robert Hunt explains that determining the effect that will be stronger "depends on the initial rate of innovation in that industry, which, in turn, depends on the opportunities for technological improvement and the resources devoted to perfecting those improvements".

In an industry where innovation is slow, patentable inventions are likely to have a high value because it will take a lot of time for other firms to generate new technologies that provide a competitive advantage. If a patent is easier to obtain, this improves the probability that the firm will have a patent that is valuable for a long time. It is expected that the other firms will take time to make even marginally competitive discoveries due to the slow rate of innovation. Since profits far in the future have a much lower value than current profits, this will not significantly decrease the value of the patent. Consequently, increasing the chances of obtaining a patent will most likely have a more significant effect than the reduction of the value of the patent on the expected returns to research and development activity. This will result in an increase in research and development activity.

In an industry where innovation occurs much more rapidly, roughly the opposite is true. New inventions are produced more frequently and will compete with existing technologies sooner. A patent is typically less valuable in this type of industry because a given invention will generate fewer profits over less time. Additionally, increasing the probability a competing firm can get a patent increases the probability that that firm can become the market leader, which also reduces the value of a given patent for a firm. Here the reduction of the value of the patent will most likely have a more significant effect than increasing the chances of obtaining a patent on the expected returns to research and development activity. This will result in a decrease in research and development activity.

There has been, without question, a marked increase in patenting over the past two decades. There are several general possibilities that could account for this increase. On one side, a larger proportion of innovations could have been patented. Alternatively, inventors could be making discoveries at a much higher pace. Possibly, if not probably, it was a mix of both. If patenting activity increased within the United States during this period of time, but did not increase abroad, then Robert Hunt surmises that "the relaxation of patentability criteria in the 1980s" is a likely explanation. He goes on to say that increased patent activity in the United States by foreign inventors would reinforce this hypothesis. This would suggest that a larger percentage of inventions are being patented. If the increase is, by and large, uniform in the United States and internationally, then it is more likely that increased technological opportunities are affording more innovations.

Patenting trends in the United States, Europe, and Japan were examined by Samuel Kortum and Josh Lerner. They discovered that European inventors did increase their rate of patenting in the United States during the latter part of the 1980s; however, this trend did not continue through the 1990s. During the 1980s, inventors from Japan "significantly increased their patenting activity, both at home and abroad". The authors believe this is a continuation of a trend that started in the 1960s. American inventors increased their level of patenting within the United States, but they also increased their level internationally. It is these authors' conclusion that the increase is far more likely explained by an increase in opportunities in the technological fields than by "a change in the treatment of patents" by the American court system.

Additionally during this study, Samuel Kortum and Josh Lerner examined any evidence of an alteration in the value of patents. They found that renewal rates declined during the early part of the 1990s, indicating a reduction in value. This evidence can be seen as supporting the theory that relaxing patentability standards "caused the profits earned on patents to erode more quickly". However, it is quickly pointed out that renewal rates declined internationally, as well, making this interpretation questionable.

There are possibly other ways to measure whether making patents easier to obtain have decreased their value, affecting expected returns for research and development. Bronwyn Hall investigated the stock market's valuation of firms' research and development investments. He found that roughly one thousand publicly traded companies increased their investments in research and development, according to market valuations, through the majority of the 1970s, but that there was a reduction in investments beginning after 1983. Hall found this effect most evident in the electrical and computing industries. He concluded this was due to "more rapid technological obsolescence and the competitive effects of entry by new firms". At the beginning of the 1990s, the companies returned to increasing investments in research and development. Robert Hunt concludes that the behaviors found by Hall cannot be attributed to a single factor, but they are hardly supportive of the theory that relaxation of patenting criteria amplified the market value of research and development investments during the 1980s.

Robert Hunt conducted a study of the market valuation of research and development investments made between 1976 and 1994 by a dozen American semiconductor firms. Looking at only a particular company's investments, there was substantial increase in the market value; however, this started after 1989. While it is possible that this is due to the relaxation of patenting standards, it is unlikely since this is roughly five years after the relaxation occurred. Hunt continues to say that within the semiconductor industry, the research and development efforts of a company's rivals are also very relevant. Due to the prevalence of reverse-engineering within that sector, it is very likely that companies gain much knowledge from the innovations of others; of course, these innovations are related to the rivals' research and development activities. This means that spillover effects are probable, i.e. "the value of a company's own research might be affected by the research conducted by its rivals".

Three types of effects that research and development activity might have on a company's market value were studied by Robert Hunt in 1996, focusing on a dozen American semiconductor firms. These three effects were a direct effect, a competitive effect, and a spillover effect, measured by a firm's own investment in research and development, that of its rivals, and the interaction between the two, respectively. A change was found in these relationships during the 1980s by using statistical techniques. During the first part of the decade, research and development by a rival would decrease the valuation of a firm. This is the competitive effect. The portion of a firm's market value due to the direct effect was concluded to be quite small, but it would increase the more the rival firms invested in research and development, indicating a positive spillover effect. Hunt believes that there is no unique theory to explain this, but it is consistent with an environment that is conducive to reverse-engineering.

Towards the end of the 1980s, or possibly during the beginning of the 1990s, there were marked changes. Research and development activity by a rival no longer caused a reduction in a firm's market valuation. In some instances, it would actually increase the valuation. Simultaneously, the direct effect had a notable increase in magnitude. Interestingly enough though, investments in research and development made by rivals decreased the value of research and development of a firm, indicating the spillover effect was now negative. Hunt concludes that this is "consistent with a shift from an environment of significant reverse-engineering to one relying more heavily on patent protection". He further suggests that the change in the competitive effect is indicative of a move from a market of direct competition to one where firms typically use components from other firms in their products. Finally, he believes that the reversal in the spillover effect is possibly due to the increased ability of a firm to preclude rivals from using its technologies through the use of patents. The aggregate effect of these changes is that "the market value of R&D investments for this group of semiconductor companies during the late 1980s and early 1990s was either the same as or lower than it was in the early 1980s". This can hardly be viewed as supportive of the proposition that the relaxing of patenting criteria would result in increased research and development activity.

More Evidence Patents Aren't Helping

Possibly one of the most unexpected impediments to research on software patents is the lack of an official definition or category for software patents. For the most part this is not a big issue: whether a patent is a software patent is, by and large, common sense. Where this becomes an issue is any attempt to do a comprehensive analysis of existing patents. There are so many that a person could not sort through all of them; they need to be distinguished by an automated process. This usually involves patent categories mixed with keywords. Inevitably there will be some error, both of inclusion and exclusion. Fortunately, the error can be statistically quantified and does not affect the qualitative conclusions. Additionally, it simply does not factor in on theoretical analysis of the economics behind patent incentives.

Surprisingly, analysis of software patents show that very few of them are applied for by what are traditionally thought of as software companies. One study placed the figure at about five percent belong to software companies. The majority of software patents are acquired by large manufacturing firms that belong to industries known for strategic patenting. There is anecdotal evidence that software companies are patenting more in recent years, but it is not likely to change the percentages significantly.

There has been a significant increase in software patents over the past twenty years. Despite the large increases in productivity, employment of computer programmers, and investment in research and development, none of these fully account for the phenomenal growth in software patents, which have increased roughly twenty-fold. A study by James Bessen and Robert M. Hunt finds that the increase in the propensity to acquire software patents is consistent with the increase of their cost effectiveness, which occurred due to changes in patent law during this period of time. They also conclude that these patents tend to substitute for research and development in firms, reducing the intensity of research and development.

This is clearly contrary to the traditional incentive theory of patents and tends to occur primarily in industries that are inclined to strategically patent. They continue that for their conclusion to be incorrect, the following two coincidences must occur. First, the increased patenting must be due to a sizable increase in the productivity of research and development that occurs in only a handful of industries, excluding the software industry, and "yet without regard to the hardware/software distinction". Additionally, demand for research and development must be "price inelastic in those same industries, but not in the rest of the economy". They conclude that this is rather unlikely.

Even More Evidence Patents Aren't Helping

Robert M. Hunt measured the research intensity of publicly traded companies in the software and data processing sector by examining the ratio of research and development spending to sales. He surmises that a ratio that is high is indicative of a large potential for new products. He found that the ratio had increased from approximately five percent during the early 1980s to about seven and a half percent at the end of the 1990s. However, most of the increase happened before the 1990s, when the patent system was still undergoing a significant policy shift with regards to software patents, resulting in much uncertainty about software patents. He concludes that, due to this uncertainty, it is unlikely that patent policy changes could be the primary cause of the increase in research intensity. Additionally, the increase in research intensity for this sector is within a few percentage points of the increase in research intensity for all publicly held companies for these two decades.

Patents Might Lack Another Expected Benefit

In a 2004 paper, James Bessen refutes the presumption that the probability of diffusion of information is necessarily greater with a patent system. Additionally, the "market for technology" is not necessarily any greater, either. He explains that "only unconcealable inventions are patented," which means that the information a patent forces to be revealed was likely to have been learned regardless. Conversely, a concealable invention will remain as such. He does concede that the rate of diffusion of information is possibly more rapid with a patent system than without for patentable inventions, if licensing is permitted and used. Without licensing, firms will use patents to reduce, or possibly eliminate, imitation. Consequently, there will be slower diffusion of information in a patent system that lacks licensing than if there were no patents at all.

For inventions that are concealable, a patent system with licensing does have some distinct benefits. Without a patent, a firm would have to rely on trade secrets to protect their inventions. Unfortunately, it is relatively difficult to license trade secrets, even if it is in the firm's best interest. This is due to two primary issues: verifiability and expropriation. The other will firm will want to verify that what they are contracting to receive is "good" and worth the cost. However, the firm will generally not be able to reveal enough information about the invention until the other firm is under contract already or else the other firm will be able to use that information "for free".

James Bessen continues on to suggest that the market for licenses has the possibility of being larger in a system lacking patents. This is because licensing typically occurs where there is the probability of imitation, which will not necessarily be direct copying. Since patents serve to restrict the probability of imitation, the result will be less need to license. Bessen concludes that, given that firms have the choice to use patents or trade secrecy to protect an invention, his findings suggest that the ability to patent will not increase the diffusion of technical information, and actually has the possibility to retard it.

Can Patents Slow Innovation?

Some key characteristics of the software industry make it susceptible to another issue exacerbated by the increasing propensity to patent. With sequential innovation, subsequent inventors will be subject to claims by prior inventors. Expecting the cost of these claims, the subsequent inventors may perform a sub-optimal level of research and development. Innovation in software is very much sequential. This problem is known as holdup. Licensing prior to performing research and development, or *ex ante* licensing, can avoid holdup.

Unfortunately, *ex ante* licensing will generally only occur when information is symmetric, which is rarely the case. Typically, sequential innovation occurs because the second firm has specialized (i.e. private) information. The second firm will not want to reveal it and give up its advantage. The first firm, who has the patent, will not know the cost of using this so it will not offer an optimal license. It may be possible that shorter patent terms can mitigate this effect. It is unclear what other effects may occur, though. Analysis by James Bessen concludes that in industries where inventing around a patent is feasible and/or the outcome of litigation is uncertain, there will be little *ex ante* licensing. Empirical evidence in the software industry bears this out.

Strategic Patenting Introduces Another Wrinkle

Traditional patent theory assumes that a product will have, at most, one patent covering it. This does not hold up with complex technologies. Typically, in this case, multiple patents are awarded to multiple parties during the commercialization of a product. If criteria for acquiring a patent are lax, firms tend to acquire "thickets" of patents. This is especially true for "incumbent firms in mature industries". When asserted aggressively, cross-licensing will cause profits to be shared, resulting in sub-optimal incentives to invest in research and development, according to research by James Bessen. This is because "cross-licensing sharply reduces the incentive effect of lead time advantages because the winner's profits are included in the bargaining over a cross-license and are shared in the bargaining solution".

If lead time advantages remain high, and with strict patenting standards, firms will rather tend to enact strategies of "mutual non-aggression". This results in higher research and development incentives. If the number of patent-holders a firm must negotiate with is high, the firm will likely face excessive transaction costs, among other problems. While patent pools have been suggested as a solution for this, the infrequency with which they have occurred suggest there are other issues with this approach. There is empirical evidence that firms generally consider lead time advantage to be more valuable than patents, except in the pharmaceutical and chemical industries. Under common conditions, an increase in patenting standards decreases the likelihood that cross-licensing occurs and increases the likelihood that mutual non-aggression occurs. The converse is also true.

Ironically, patent thickets tend to be an example of the actions of firms counteracting public policy. Firms tend to create portfolios of patents staggered over time to achieve the effect of long term patents when the government establishes a short patent term. Additionally, firms tend to create large portfolios of related patents to achieve the effect of broad patents when the government establishes a narrow patent scope. James Bessen concludes that many firms pursuing strategies involving patent thickets will result in an equilibrium where there is less overall investment in research and development. The patents end up substituting for research and development. With lax patenting criteria and complex technology, he finds that "patents serve to subsidize the losers of innovation races (paid by the winners), especially if those losers are large patent holders in mature industries". Additionally, patent thickets may provide barriers to entry in a market because they force the new entrant to quickly develop their own patent thicket.

* * *

While the research is illuminating, the changes they point to tend to be, unfortunately, more descriptive than prescriptive in nature. One theme is fairly conclusive: the characteristics of industries involved with information technology cause patents to affect the behavior in these industries differently than in more traditional industries. The assumption that imitation is bad may no longer be true. Nor are patents necessarily an incentive to conduct research and development.

IV. Proposals for Change

Given the software industry's relative youth and explosive rate of growth, it's not surprising that worldwide standards of software intellectual property protection are still being perfected. There are numerous interested players (corporations, consumers, society), and a not inconsequential amount of academic research targets the topic. Grievances with the current system are numerous, and so are the proposed solutions and alternatives.

From the field of existing proposals, we see two distinct groups: *incremental* proposals, which improve the quality of patents issued, but preserve the structure and protections granted intact; and *structural* proposals, which modify the actual definition of a software patent (or which call for ending software patents altogether).

It is interesting to note that in general, the incremental proposals we encountered were proposed by members of the software industry, who stand to gain the most from the improved patents that would result from those incremental proposals. Most of the structural proposals, on the other hand, came from academic research and public interest groups, which in general do not *directly* hold and benefit from software patents. Even if software patents always serve the interests of society—an assertion in some amount of dispute—it's clear that they serve different parts of society differently.

In this section, we survey both incremental and structural proposals from various groups, as well as one idea of our own.

Incremental Proposals

In January and February of 1994, the U.S. patent office held hearings regarding the patentability of software. The purpose of the hearings was to standardize the rules upon which software inventions could be afforded intellectual property protection, which up until that point had been determined through various individual case law decisions. Numerous large software companies were invited to present their positions regarding software patents. While most companies voiced either strong support or dissent for software patents, few provided solid details or suggestions towards improvement or alternatives. One exception was the statement prepared by Oracle Corporation which, while opposed to software patents in general, detailed a targeted list of changes to software patents to improve their effectiveness. The proposed changes were:

Improve prior art research.

Prior art research is the investigation of existing granted patents to verify that an invention under patent consideration is in fact new, novel, and non-obvious. The unique difficulty introduced by software patents is that the existing capabilities of prior art research are insufficient. Without proper analytical capacity and classifications, research is slowed and accuracy suffers, resulting in increased duplication, overlap, and frivolous patents. To remedy this, the Patent and Trademark Office could standardize on a format for describing and categorizing granted software patents at a more granular level. This new system would allow new software patent applications to be properly categorized and cross-referenced against the database of existing patents. With this improved capacity for prior art research, the general quality of software patents would increase, and granted patents would be stronger as more frivolous patent applications are weeded out. Over time this could lead to a decrease in patent applications from the software industry as the rate of rejection rises, and resources that were previously used towards patent applications could possibly be used towards further innovation.

Reduce the patent review process to six months.

It currently takes two to three years for a patent application to be evaluated and approved or rejected. In the dynamic software industry, the rapid pace of innovation and the speed with which innovations are incorporated into products means that most software "inventions" are either widely used or obsolete by the time the patent is granted. To address this, the process by which patent applications are evaluated would have to be sharply streamlined. With quicker turnover on patent applications, resources would be freed and applicants could be compelled to take advantage of the shortened wait times by filing more applications, including applications for frivolous innovations.

In spite of this, we have not encountered anything suggesting that applicants, given shorter application processing times, would actually divert freed resources to further research and development.

Hire and retain more qualified software patent examiners.

Not only are the current prior art tools not up to the task of validating newness, novelty, and non-obviousness, but the examiners who perform the research are also not sufficiently qualified. Unqualified examiners increase both the time taken to process patent applications, and the number of frivolous applications that are approved. Unacceptable levels of attrition further compound the issue. Implementing this proposal would involve improving the training system and curriculum so examiners are well-versed in the nature of software inventions and the state of the art, and better compensation rates in line with the industry, to better retain those qualified examiners.

Establish, in conjunction with industry, committees to determine proper standards of novelty and non-obviousness pertaining to software inventions.

Even with better prior art research capabilities and more qualified examiners with whom to perform the research, not having proper standards of novelty and non-obviousness by which to judge software patent applications would lead to confusion, delay, and ultimately more duplicate patents. This proposal suggests the involvement of those entities who work most closely with software innovations, software companies, be directly involved in determining the standards of novelty for software patents. While at first glance, one might suspect a strong conflict of interest in having those who would apply and benefit from patents be privy to setting the rules by which those patents are granted, the numerous companies contributing input are in fact competitors with one another, and are thus motivated to suggest fair standards for evaluating the novelty of software patents. We would expect there to be a fair amount of overhead involved with the creation of these committees, as well as the periodic cost of reconvening as necessary to address previously unforeseen software novelty issues. Also, there would remain a question of fairness as to which companies are allowed to participate in the joint committees, and whether their interests would be promoted at the expense of likely smaller companies that would have no say in the matter.

* * *

All of these changes relate to improving the process by which patent applications are evaluated. While they appear to be fairly simple, common sense suggestions, their effective implementation is hardly trivial. The dynamism of the software field continually complicates attempts to find prior art. Attempting to quarter the average time of any government approval process is a daunting goal. There would be large budgetary considerations associated with the training, hiring, and increased compensation of qualified examiners. And the overhead associated with creating and operating joint PTO-industry committees would be tremendous.

Assuming for the moment that these hurdles proved surmountable, what would the landscape look like with these changes in place? Software patents would probably provide an even greater amount of protection than they do today, because they will be recognized to be of higher quality. The process by which they were approved would be greatly streamlined, which benefits both existing patent holders and future applicants.

Given the research available to us, however, there is nothing in these proposed changes that would lead us to believe that the overall rate of innovation (possibly measured by research and development dollars spent) in the software industry would be boosted by this set of changes. Furthermore, implementation of these proposals would be insufficient to address the numerous other issues raised in prior sections, such as the stalling of sequential innovation in the presence of an innovation monopoly.

Structural proposals

The improvements we detailed turn out to be popular among players who have an interest in higher quality patents: those who possess patents and work hard at invention, as well as (most probably) society as a whole.

Fewer players choose to tackle the question of whether society might be better off if the patent *protections* themselves were to be severely curtailed, or even eliminated. Economists have, however, tackled this very issue. And if software patents aren't accomplishing for software what the founding fathers had in mind when they saw the need to protect intellectual property, then it may be high time to change the patent protections themselves, or even eliminate them.

In actually considering the reduction of patent protection, there seem to be two variables which have evolved over time and vary from jurisdiction to jurisdiction.

First, we can adjust the *length* of patent protection. A reduction of protection length for software would continue to protect ideas, but reduce the innovation-inhibiting effects documented earlier. Securing a patent has its rewards, but becomes less of its *own* reward.

Second, we can adjust the *breadth* of the protection. For example, if a company produces a piece of software, we must decide how different a competing product must be in order to avoid infringement. We must also decide how "expensive" infringement is—from the lenient royalty payments to the strict payment of triple damages.

Reducing protection length

With respect to protection length, we can summon some intuition to explore its effect on the rate of innovation. If patent protection is very short, we increase the "frequency" on innovation: that is, companies will seek to more quickly produce inventions that can be patented, to replenish their portfolio which contains aging (and expiring) patents. This effect sounds positive, but is counterbalanced by a decrease in the "size" of innovation: the bets are likely to be smaller, since results need to be patentable within a shorter period of time and will end up receiving shorter protection. Frequency and size are *both* elements contributing to the overall *rate* of innovation. Models need to weigh both frequency and size, and maximize for the overall rate. When the House

Committee on Science and Technology reviewed patent policy in 1988, it focused on a policy that would optimize for the overall rate of innovation.

Numerous academic works, however, paint an even more complicated picture. William Nordhaus, in his 1969 *Invention, Growth and Welfare*, was one of the first to point out that the maximal overall rate of innovation may not correspond to maximal societal welfare. We benefit not when innovation and competition are at its peak, but rather when an optimal innovation *size* is achieved—his tradeoffs are around how big the *bets* should be.

More recently, work by Andrew Horowitz and Edwin Lai (in their 1996 "Patent Length and the Rate of Innovation") has concluded that the length that maximizes the overall rate of innovation turns out to be *longer* than that which maximizes societal welfare.

In the end, at least as it applies to software, perhaps these academic studies are precisely that—academic. As companies repeatedly pointed out at the U.S. patent office hearings in 1994, software technology advances at a rate that makes a mockery of the time it takes merely to secure approval. The creeping obsolescence of protected technology makes a formal reduction of the protection period seem at best a superfluous formality.

Reducing protection breadth

With respect to protection breadth, we again summon some intuition to explore its effect on innovation. A "wider" patent, one that requires more differentiation to avoid infringement, clarifies choices for consumers between the patented product and its unpatented competitors. This permits higher prices to be charged, but this is balanced by the (presumably desired) innovation that a "wider" patent incentivizes. A "narrow" patent, on the other hand, muddles choices for consumers and therefore forces prices down, with a corresponding depressive incentive to the patent holder.*

In Paul Klemperer's famous 1990 analysis "How Broad Should the Scope of Patent Protection Be?", the trade-off between patent length and breadth is studied. He finds that when all consumers have identical transport costs—which is theoretically (though not actually, with network effects) the case with software—then patents of infinitesimal width and infinite life seem to be socially optimal. He states the intuition best:

> ...when all consumers have the same transport costs, they all have the same preference between the patentholder's product and the competitively supplied variant, so the patentholder sets a price such that non substitute the competitively supplied variant... Since narrower patents constrain the patentholder to lower prices,

* The breadth of patent protection has changed over time and differs between countries. Eli Whitney, he of the famous cotton gin patent, had the misfortune of securing it at a time when patent breadth was minimal. Competitors simple filed almost identical applications and successfully received cotton gin patents of their own. We remember Eli Whitney and not his imitators, but he actually made very little money from the patents themselves.

> the narrowest possible patent is therefore the most socially cost-effective way of awarding profits to an innovator.

Though he describes scenarios under which wider patents should be considered, they don't map well to the world of software.

It is interesting to take Klemperer's suggestion and combine it with our previous point about the rapid obsolescence of software patents: the distinction between infinitesimally narrow patents that become quickly obsolete, and no patents at all, seems small indeed.

Patents with mandatory licensing

Another way to narrow the protections afforded by patents is to make sharing of the invention somehow mandatory; that is, the inventor, in exchange for exclusive rights, is required to make the invention accessible for development via licensing for a fair price. Though this idea may not be original, we haven't seen this idea discussed extensively anywhere else, and here we will evaluate some of the merits of this solution.

The obvious benefit of this solution is that it allows for competitors to use the invention to produce useful products, while still rewarding the patent holder via a guaranteed income amounting to a rent. Companies will be less afraid to develop a project that may be infringing, because the consequences of infringement are capped at a reasonable royalty payment. As pointed out in earlier analysis, in software it's frequently the case that the inventor actually benefits from a competitive ecosystem, and may choose to waive licensing fees altogether; that is fine. What we propose is simply that in exchange for patenting an invention, the inventor gets a reasonable, guaranteed income stream, but that access to the invention is also guaranteed at a reasonable price.

On closer analysis, though, this solution has numerous problems of its own.

Firstly, it's difficult to decide what a "reasonable" royalty is. Some patents may be more difficult to develop than others, and therefore less worthwhile to pursue if the rent is price controlled.

Secondly, it's difficult to extend this model to multiple generations of improvements. An original patent holder is compelled to license to the second generation innovator, but what next? If a third generation innovator enters the market, does this innovator pay royalties to both the first and second generations? If so, at what point do we make future generations prohibitively expensive? What incentive remains for the first generation innovator if they can sit back and collect rents from multiple generations?

Thirdly, how do we define a "generation" of improvement, in practice? Does the licensee have to show actual improvement to the invention? What if their improvement is just offering the invention at a lower price? At that point, it sounds like a pretty sweet deal for the imitator and a bum deal for the inventor (though, again, there's some evidence that the inventor himself, spurred by the competition, will actually stand to gain more than just sitting pretty collecting rent).

In the final analysis, it seems that any solution in which either the Patent Office or the courts mandate that inventions must be accessible at a certain cost is not truly tenable. One thing at which markets excel—even if they sometimes fall short of optimizing for societal welfare—is unearthing the price of a commodity or an idea. Patents are a vehicle to make ideas valuable and tradable, and by granting the right to the inventor to establish royalty rates, we allow market forces to sort it out. If the prices are mandated, even with the good intentions of accelerating innovation and maximizing welfare, they can trigger the counterproductive results we have outlined above.

The big hammer: elimination of software patents

Given both the amount and intensity of interest from all sides on the issue of intellectual property rights for software, it is only fair that we consider the full spectrum of changes that have been proposed. The most extreme proposal that we have encountered is to eliminate software patents altogether. Surprisingly (perhaps), there exists a wide body of support for this idea.

In the U.S. patent office hearings on software patents in 1994, a not insignificant number of software companies declared their public support for the elimination of software patents. Chief among the reasons cited were that the elimination of software patents would be healthier for the industry as a whole, by eliminating any artificial boundaries and overhead on sequential innovation. The statement by Adobe Corporation was among the most forceful. Remarking on the "patent litigation tax" wrought upon the industry by software patents, Adobe's representative remarked:

> This state of affairs might be acceptable if there were a corresponding benefit for patents in the software industry. However, I see none. Companies that have trumpeted their fundamental software patents are not leaders in software innovation. Conferring monopoly positions in an industry that was already the most innovative of all will promote stagnation rather than increased innovation. When companies turn from competing by offering the best products to earning money by the threat of patent litigation, we will see our best hope for job creation in this country disappear. An industry that still generates tremendous job growth through the start-ups of two guys in a garage will not continue to grow when a room for a third person, a patent attorney, needs to be made in that garage.

Public support for the idea of outlawing software patents is not unheard of in the U.S., but the strongest base of public support for the idea appears to be in the EU, where patent law regarding software is still undergoing rapid change and close examination. While as stated previously the EU's written law forbids the patentability of computer programs, the trend through recent applications and litigation is the relaxing of this restriction. As such, groups such as The Foundation for a Free Information Infrastructure have been increasingly vocal about keeping the EU standards strict. The position taken by these public groups in support of disallowing software patents is not based solely on what is best for software in general, but is also mixed with some nationalistic and anti-corporate leanings, as noted in the position of the website nosoftwarepatents.com:

> Without software patents, Europe could save costs, foster innovation, enhance security, and create jobs. Thanks to Linux and other open-source software, Europe has the chance to gain independence from Microsoft and other large American companies. However, if the EU allows software patents, then that's the beginning of the end for Linux. Not only for Linux. It's just a prominent example.

The strongest voice in favor of eliminating software patents, based on the existence of supporting evidence, comes from recent economic research. As noted previously, economists such as Bessen and Maskin have studied numerous models of the effectiveness of patents in the software industry. The main points from such research supporting the elimination of software patents are:

The software industry grows through dynamic innovation. Innovations in software are sequential, and each new innovation builds upon existing and previous innovations. In industries with dynamic innovation, the elimination of barriers between innovations results in a higher rate of innovation in the industry as a whole.

The amount of research and development in the industry has not increased since software patents started being granted. Since the main argument for patentability is to foster and promote innovation, it would be expected that research and development in an industry that would benefit from patent protection would increase if patents were allowed. Bessen and Maskin's research showed this is not so, that the percentage of capital devoted to research and development in the software industry has not appreciably increased in the period where software patents have been allowed. This is interpreted to mean that the patentability of software has not increased the amount of innovation in the software industry.

The software industry has adopted ways in which to work around software patents. Rather than using patents as intended, to protect their own innovations, software companies have gone about acquiring "patent thickets" to use as bargaining chips to gain access to other companies' innovations. The actual amount of infringement litigation is surprisingly small. This shows that companies are approaching patents as but a hurdle or obstacle around which to work, and that software patents are not serving their original intent.

* * *

Having made these arguments, it's worth pointing out that ending software patents is not a trivial matter, both practically and even theoretically.

Practically, it is the case that any computer program, through fairly mechanical manipulations, can be represented equivalently as a mathematical algorithm (which the courts have ruled *cannot* be patented) or as the manufacturing process for hardware (which courts have ruled *can* be patented). A common sense standard for what constitutes software may well fail if there becomes an incentive to change how the invention is represented.

Theoretically, the road is equally murky. Patents, as we recall, were originally conceived to protect *ideas* themselves. It challenges the policymaker to delineate which ideas deserve protection and which do not. Even if fairness and consistency are sacrificed from consideration, it is still quite difficult to judge which set of ideas can best be nurtured through the promise of protection (the drug industry, obviously so), and which through open season of competition (the software industry, perhaps so). As software begins to pervade all industries—one could make a very reasonable case that today, drug companies *are* software companies, producing products with a biochemical encoding—the questions become ever more difficult to answer.

References

Bessen, James. March 2004. "Patents and the Diffusion of Technical Information".

Bessen, James. 2004. "Holdup and Licensing of Cumulative Innovations with Private Information", Economics Letters, v. 82, no. 3, pp. 321-26.

Bessen, James. "Patent Thickets: Strategic Patenting of Complex Technologies".

Bessen, James and Eric Maskin. 2004. "Intellectual Property on the Internet: What's Wrong with Conventional Wisdom?"

Bessen, James and Eric Maskin. July 2002. "Sequential Innovation, Patents and Imitation", MIT Working Paper 00-01.

Bessen, James and Robert M. Hunt. March 2004. "An Empirical Look At Software Patents", working paper no. 03-17/R.

Horowitz, Andrew W. and Edwin Lai. November 1996. "Patent Length and the Rate of Innovation."

Hunt, Robert M. Q1 2001. "You Can Patent That? Are Patents on Computer Programs and Business Methods Good for the New Economy?" Business Review.

Hunt, Robert. November/December 1999. "Patent Reform: A Mixed Blessing For the U.S. Economy?" Business Review.

Klemperer, Paul. Spring 1990. "How Broad Should the Scope of Patent Protection Be?"

Nordhaus, William. 1969. *Invention, Growth and Welfare.*