

HE7601  
.S46  
1989 i

COMMUNICATIONS  
FORUM

SOFTWARE PATENTS:  
A HORRIBLE MISTAKE?  
March 23, 1989  
Seminar Notes

**MASSACHUSETTS INSTITUTE OF TECHNOLOGY  
COMMUNICATIONS FORUM**

**SOFTWARE PATENTS:  
A HORRIBLE MISTAKE?**

**March 23, 1989**

**Seminar Notes**

**Daniel Bricklin, Software Garden, Inc.  
Stephen D. Kahn, Weil, Gotshal & Manges  
Lindsey Kiang, Digital Equipment Corp.  
Robert Merges, Boston University School of Law  
Pamela Samuelson, University of Pittsburgh School of Law  
R. Duff Thompson, WordPerfect Corp.**

**Brian Kahin, Esq. (session moderator)**

**Gail Kosloff, MIT, Rapporteur**

## MASSACHUSETTS INSTITUTE OF TECHNOLOGY

### COMMUNICATIONS FORUM

This session focused on the dynamic microcomputer software industry and what impact patents might play in the future of this industry. The speakers also addressed the appropriateness of applying either copyright or patent law to emerging micro software products. Lessons were also drawn from other industries as to what tack might be the most suitable for the software industry. Numerous questions were also raised about the tradeoffs involved in choosing to protect or not protect software, including the long-term impact on creativity and investment.

Brian Kahin, attorney and former Coordinator of the MIT Communications Forum, opened the session by raising some of the concerns surrounding the topic of patents and microcomputer software. He referred to an article that appeared the previous week in the Wall Street Journal which was comprehensive in addressing this topic. Kahin noted three areas of particular interest: (1.) there seems to be an unprecedented movement from a copyright environment to a patent environment; (2.) the software industry is special in nature in that it has experienced explosive growth and prolific production of programs; (3.) the special nature of software as an "information" product.

Kahin explained that the session would cover several sets of issues related to the protection of software as "intellectual property," including: (1.) the fact that patents protect ideas at a relatively high level of abstraction as compared to copyright; (2.) the structure of the patent regime as differentiated from the copyright regime since patents provide an "absolute" monopoly for the inventor, while copyright provides protection against copying; (3.) the difference between the novelty standard of patent and the originality standard of copyright; (4.) the complexity of patents as opposed to copyright; (5.) the costs involved in searching, applying for, and litigating patents ("the sport of kings"); (6.) issues of waste and inefficiency; (7.) the problem of inconsistent standards; (8.) the system's bias toward large corporate enterprise; (9.) problems involving the inability of the Patent Office to classify software and to evaluate novelty.

Kahin raised concerns about the appropriateness of patents in the software industry. He believes as "we are moving into a new

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

COMMUNICATIONS FORUM

publications environment in which computers create a new way of managing and digesting information, e.g., hypertext, that patents may start to come in conflict with values about information and freedom of speech."

The first panelist, Stephen Kahn, is a partner in the New York law firm of Weil, Gotshal & Manges, and a graduate of Yale Law School. Kahn described the background of software patents and traced the history in the United States of how patents came to apply to software-related inventions, from the nineteenth century to the present.

Kahn pointed out that a patent is a "negative" right: the right to keep someone else from making, using or selling the subject matter of the patent without the owner's permission for a term of 17 years. He noted that, since a patent is a negative right, the owner of a patent is not necessarily free to make, use or sell its subject matter. Paradoxical as it may seem, Kahn pointed out that the acts of making, using or selling by the patent owner may infringe on someone else's patent.

He went on to discuss how patents are more powerful than and have greater potential impact than copyrights and trade secrets. Kahn remarked that even if you develop your idea independently and without knowledge that a similar idea had been developed by another company, you may infringe on that other company's patent, and your ignorance is not a defense. For example, if you were to use a certain windowing technique in your software, you might infringe on a patent AT&T owns on such a technique; your lack of knowledge of AT&T's patent would not protect you from being sued and being found guilty of patent infringement. Also, Kahn noted that the absence of secrecy of the protected subject matter is not a defense to patent infringement. Indeed, once a patent is issued "the subject matter necessarily becomes known."

Kahn then compared patent and copyright protection. He noted that a patent is much broader in scope than a copyright and is not limited to any particular structure or organization. According to Kahn if a patent lawyer does his/her job carefully, a patent claim will be broad and "if drawn to a method, will cover several alternative ways of performing the method, or, if drawn to a machine or system, will cover several alternative structures for the machine or system." Kahn stated his belief

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

COMMUNICATIONS FORUM

that patents have a much greater potential scope than either copyrights or trade secrets in the software industry.

On the otherhand, Kahn cited some factors which, in his view, reduce the potential impact of software patents. Foremost of these, he said, is the "obviousness standard." Under the obviousness test, the Patent and Trademark Office must find that the claimed invention would not have been obvious, at the time it was actually made, to a hypothetical person who had the average level of skill in the area to which the invention relates and who knew all the prior art that existed at that time. Kahn contrasted this with copyright law where only minimal novelty is required and to trade secret law where the question is confidentiality. He also believes it is less difficult (but still relatively difficult) to meet this test today than it was ten years ago, because of a number of judicial developments.

Secondly, Kahn believes another important issue relating to software patents today is the availability of information about existing patents. Although patent searches can reduce the risk of infringing, they are costly. He noted that one benefit of the patent system is supposed to be the incentive it gives people to "design around" existing patents and thereby discover new knowledge, but noted that a major problem with the existing system is the difficulty of locating software patents.

In the historical portion of his talk, Kahn began by noting that the United States Patent Act provides that patents shall only be granted for a "process, machine, manufacture or composition of matter, or any improvement in any of these." According to Kahn, the key issues regarding the patenting of software-related inventions have revolved around the question of whether a particular invention is a "process" or a "machine." Addressing these issues, he noted, has been a challenge for the courts.

Kahn believes that as late as the 1960s, no patent claim directed to an invention in the software area had any chance of being allowed. One reason was the doctrine called "mental step" which held that methods of calculation involving mental operations, e.g., methods of computation, were not patentable. A second barrier was the doctrine known as "function of a machine" which would not allow a patent for a claimed process that

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

COMMUNICATIONS FORUM

described the function of a machine, even though a claim to the machine might be patentable. Kahn also noted that 20 years ago the term "process" in the Patent Act would not have included a computer process since that type of process did not result in the physical transformation of an object. Kahn noted that, as computer programming became a significant enterprise in the late 1960s, these barriers began to crumble.

Kahn then provided a historical overview of the legal system as it relates to software patents, noting the roles of the Patent and Trademark Office (PTO) and the Court of Appeals for the Federal Circuit (CAFC; formerly the Court of Customs and Patent Appeals). If a patent application is rejected by the examiner within PTO, the applicant can appeal that rejection up through the PTO and then to the CAFC. That court, Kahn indicated, differs from other federal circuit courts of appeal in that several of the judges, and many of their law clerks, have a technical education and some background in patent law. Kahn said that, in several cases involving computer software-related patent applications beginning in the late 1960s, the PTO rejected the applications but the CAFC overturned the rejections and ordered the PTO to issue patents. In those cases, the CAFC threw out the "mental step" and "function of machine" doctrines.

Kahn indicated that the Supreme Court did not at first agree with the CAFC's easing of the law towards software patents and occasionally stepped in and reversed the CAFC, siding instead with the PTO. This "tug of war" between the Supreme Court and the CAFC, Kahn said, continued throughout the 1970s until the case of "Diamond v. Diehr," 450 U.S. 175, was decided in 1981. In this case, Kahn explained, a 5-4 majority of the Supreme Court decided that Diehr's process (which involved a rubber molding press operated with the aid of a digital computer) was patentable even though several steps of the claimed process involved use of a mathematical formula implemented in a digital computer and though use of the algorithm was said to be an important part of the process. Kahn added that, for a few years after "Diehr" the CAFC refined the law according to its view and then, after a case called "in re Abele" was decided in 1983, there was silence in the courts on the issue of the patentability of software-related inventions. Kahn called that "the silence before the storm."

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

COMMUNICATIONS FORUM

According to Kahn, it became clear after those decisions that patents were available for software-related inventions if the applications were written properly to comply with the guidelines provided by the courts. Some physical reality is required to make a claim to an algorithm "statutory," according to Kahn, though it can be as minimal as displaying a shade of gray on a computer screen.

In summary, Kahn concluded that "software patents have arrived and the legal test for statutory subject matter is now very permissive." He noted that, if any one single fact would make his point, it is that IBM, according to industry reports, is obtaining nearly 200 patents per year on software-related inventions.

The next speaker, R. Duff Thompson, is the Secretary and General Counsel for WordPerfect Corporation and a graduate of Brigham Young University Law School. He noted that it is WordPerfect's view that the use of patents will impact the software industry adversely and will make changes in ways we cannot imagine at this time. Thompson explained that he would be speaking from the perspective of the software developer in taking a look into what the future might hold for software products and patents.

Thompson reiterated that a patent is a "negative" right, as opposed to a copyright which is a "positive" right. He noted that when he was in law school it was a given that software did not fall into any class of patentable subject matter and this line of thinking has continued through the 1980s until "Diamond v. Diehr." He also agreed with the previous speakers that one major problem with today's patent system is the inability to discover the existence of a patent while it is an application.

Thompson went on to discuss a few specific patents which relate to computer features that are growing in popularity. He noted a patent issued in 1984 to Apple Computer that relates to "pull down windows" activated with a mouse which covers the process of depressing a button while the cursor is overlaying an item on a menu bar to cause a submenu to drop down. Thompson acknowledged that even though this is a patented process many companies today are doing similar processes. He raised the problem of "mine fields" in this regard. Thompson noted that the numerous software companies going into the presentation manager area

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

COMMUNICATIONS FORUM

have to be concerned with the impact of existing patents on the development of their products. Thompson believes if patents such as the one issued to Apple exist then "there are undoubtedly others in the process that are going to have at least as much impact as something of this nature."

Thompson also cited a patent that was issued on November 1985 to AT&T directed to a programming technique showing several overlapping bit-mapped displays on a computer display in which all displays are active at one time. This is another area which many software companies today are interested. He also raised the example of a 1987 patent issued to IBM to illustrate what a narrow area of concern the Patent Office is issuing patents. The IBM patent dealt with "a method for controlling the printing of documents in a wordprocessing environment so the document image appears at the same location on the paper when the document is printed on different printers having different paper positioning parameters." Thompson contrasted this with the fact that today WordPerfect has printer drivers for over 300 different kinds of printers and each one of these printer drivers has some kind of code instruction on how to center the document and text on the page using different printer parameters. Thompson questions whether the Patent Office was able to understand what "prior art" was in order to issue a patent like IBM's.

Thompson noted that WordPerfect believes that the patenting of software is the wrong approach. He believes that the Patent Office examiners are probably doing the best job they can with the information available to them, but that information is incomplete. According to Thompson a problem inherent to the software industry is that such important information (e.g., related to source code algorithms) is "secret" and is therefore not available to the patent examiners. Thompson also cited the case of Xerox and a series of patents they obtained in 1988 relating to icons. Such things have made WordPerfect's job as a vendor more difficult.

According to Thompson, WordPerfect is in a dilemma since they are philosophically opposed to software patents like the examples he mentioned earlier. WordPerfect believes the the existence of such software patents is raising the "barriers to entry" for small innovative companies or individuals who he calls the "hallmark" of the computer software industry. As a company,



MASSACHUSETTS INSTITUTE OF TECHNOLOGY

COMMUNICATIONS FORUM

WordPerfect is now examining the concept of patents and whether they will use them as a "sword or shield." He personally believes it is a waste of energy for a patent holder to spend their time in search of patent violators. He sees more of an alternative in using patents as a shield and exploring cross-licensing agreements and alliances with other companies. In closing Thompson acknowledged that the future is filled with a lot of uncertainty although it seems patents will become more the rule than the exception.

Lindsey Kiang is Senior Counsel for Digital Equipment Corporation and is a graduate of Yale Law School. He was previously General Counsel at Lotus Corporation and before that was the General Counsel for Yale University. Mr. Kiang is of the opinion, in discussions like this, patents may seem to be an isoteric subject matter. He will try to demystify some of this in today's presentation.

Kiang believes that a lot of the policy issues being raised by the subject of patents, and to some extent copyright law, have been raised at other times with other technologies. In his opinion patents are not "a horrible mistake," but are salutary for the industry. Kiang believes that first and foremost patents give an element of protection to computer programs which is not available under copyright protection since patents can protect the actual concept or idea underlying an invention. This is a "deeper and broader" protection than what can be provided by copyright law. Companies like DEC believe that patent protection is necessary to protect their investment in software R&D especially when they are spending more than \$1 billion a year on such work. He believes that R&D costs will only rise as software development becomes more advanced and complex to meet the needs of customers. Kiang believes that companies like DEC will continue to feel an obligation to its investors to protect its R&D investment with patents. He believes that patent protection for software will continue to stimulate innovation and creativity.

Kiang does not believe the issue of patents is a "big versus little company" issue. He would argue that a small company needs the power of patent protection even more than the big company. In his opinion, small companies are usually putting everything at stake on one product.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

COMMUNICATIONS FORUM

Furthermore, he does not believe the cost of searching or filing patents is prohibitive when the average search costs in the \$1,500-\$2,000 range. He noted that on average a patent filed in the U.S. costs about \$8,500 and this is just a cost of doing business. In response to the "land mine" problem mentioned earlier, Kiang believes that companies can spot some of the patents in the U.S. "pipeline" by tracking such patents which are also filed in Europe, where they are made available for public inspection sooner than in the U.S. Kiang does acknowledge that there are still risks and the cost of litigation is high.

Kiang observed that there are really very few patent law suits in the U.S. when compared to the number of patents issued in the U.S. He noted that when most disputes arise they are usually settled out of court and often are resolved by licensing. He believes this is the case in the hardware world and he would expect the trend to be the same in the software world. Kiang believes that it is an unfounded fear that patents will be too broad and licenses will not be available. He does not believe patents will stifle creativity and innovation.

Kiang ended his talk by reviewing the history of patents for computer software. He noted that the first software patent was issued to Marty Goetz of Applied Data Corporation for a program to sort large quantities of data or records in a data processing systems. Kiang stated that between 1981 and the present there have been more than one thousand software patents issued in the U.S., most in the last 4-5 years. He believes that it is therefore hard to argue that patents have stifled creativity in this industry. Kiang believes that today the software industry is very vigorous, flexible and innovative and will continue to be so despite patents or copyrights, or other intellectual property rights mechanisms.

The next speaker, Robert Merges, is an Associate Professor of Law at Boston University School of Law and specializes in the area of intellectual property law. He has testified several times in Congressional hearings and worked with Congressional staff to create patent legislation. He is also a graduate of Yale Law School. Several years ago he also worked as a technical writer for Visicorp.

Merges focused his talk on comparing the software industry to certain other industries and looking at the effects of

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

COMMUNICATIONS FORUM

intellectual property rights on these industries to get an idea of what the future might hold for the software industry. He believes that the major factor which separates the software industry from others is its "relatively small minimum efficient scale." Merges explained this to mean that a couple of people can get together and produce software that is of marketable quality.

Merges noted that there are some industries which have historically been very patent dependent, e.g., chemical and pharmaceutical industries. In contrast, some industries like automobiles and the aerospace industries have not historically been dependent on patents. He explained that the test of "patent reliance" would mean asking a company how many research projects it would undertake in the absence of patents; the patent reliant company would respond that none of the projects would be undertaken without patents. He noted that there has been a lot of empirical research in this area, especially at Yale.

Merges cited that historically patents were very important in the formative stages of several industries where they are no longer important today. He noted the automobile industry as a case in point. Merges explained how the famous Selden patent blocked a lot of progress in the early development of this industry. As a result of this an industry trade group was formed that ended up licensing the patent to any one who wanted to use it. According to Merges, the same kind of situation happened in the semiconductor industry until the Dept. of Justice brought suit against AT&T. This forced AT&T to license all their important early patents. He noted that today in the semiconductor industry companies will patent the results of their research in order "to play the game" and have future bargaining chips.

Merges also raised the problem of sometimes enforcing patents. He noted that this is especially the case with process patents since it is difficult to tell who is infringing on such patents. Merges explained that some of the broadest algorithm patents are process patents. He stressed that litigation costs often act as a disincentive to patent enforcement. He cited the Stanford decision in which a fairly low royalty rate was set for the basic DNA patent in order to avoid having to litigate everytime there was an infringement.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

COMMUNICATIONS FORUM

Merges went on to address the question of the future importance of patents in the software industry. He believes that in the short term they will probably play an important role, similar to the role patents played in the early days of the automobile industry. He does not believe that patents will cause too many problems in the software industry because "they are expensive to enforce and because it may be difficult to detect certain kinds of infringements. He acknowledged that the problem of "secret" prior art may cause future problems in the software industry. To address this problem he believes the U.S. should accept the European model which calls for publishing within eighteen months of filing. He believes that many of these problems with the patent system are generic and not specific to the software industry.

Merges went on to provide examples of industries where a companies hold on a certain patent(s) had put a deadlock on the industries growth. He noted the early radio field as an example since G.E. held the basic patents on reception and Marconi held the basic patents on transmission. In order to merge the work in these industries to develop a radio industry it required action by the Assistant Secretary of the Navy, Franklin Roosevelt. Merges called this an early form of "industrial policy." He also noted the Curtis-Wright merger, encouraged by the U.S. Dept. of the Army, as another example of government involvement to break a patent deadlock to further the development of a certain industry.

Merges acknowledged that there is alot of uncertainty in the current state of patent law and how it relates to software. He believes there will be a "spill over" effect in terms of litigation in this industry. Merges explained that "when someone takes the time to litigate, everybody who was pushing their position wins eventhough they did not have to spend any money. He therefore suggests that "having big companies litigating some of the early software cases might have some beneficial effects for the little guys."

Merges commented on the operation of the Patent Office. He believes that the office suffers from high turnover and a gap in the level of skill of the examiners as compared to the inventors. He noted that this knowledge gap is especially large in the area of biotechnology. Merges believes that this will result in the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

COMMUNICATIONS FORUM

issuance of more and more broad patents that really should not be issued. Merges feels this is an area which should be addressed.

Merges believes that even though the software industry has not been given special treatment, in terms of special intellectual property protection, that it has seen a lot of investment, competition and new products emerge. He believes that from a policy point of view there are certain micro-adjustments that the Courts and Patent Office can make to tailor protection for certain industries. Foremost, Merges believes these organizations should restrict the scope of patents such that only significant algorithms "make it through the gate." Secondly, he would like to see the scope of patents kept as narrow as possible so that the patent only covers what the researcher has actually come up with. Thirdly, Merges hopes that new legislation would be introduced to restrict, for example, the reach of algorithms, as well as provide a broad research exemption for educational research.

The next speaker, Pam Samuelson is a Professor at the University of Pittsburgh and a graduate of Yale Law School. She previously worked at a New York law firm and has done research on intellectual property protection for computer programs. She focused her presentation on the "copyright-patent interface" for computer software.

Samuelson reiterated that computer software is both patentable and copyrightable. She is very interested in the question: What about computer software is protectable by copyright and what is protectable by patent, and what is the relationship between the two kinds of law? She suggest that these are very complex areas to address.

She noted that historically patents and copyright have been two separate kinds of intellectual property law. While patents have been used for industrial technologies, e.g., machines, manufactured products, manufacturing processes, etc., copyright has been applied to expressive works like writing, painting and sculpture. She explained that in general to be "patentable" something has to be "useful" and to be copyrightable something cannot be "useful." Samuelson defined "useful" as meaning things that have a function beyond just conveying information or displaying an appearance. For example, a book is not considered to be useful in a copyright sense since it does not have a

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

COMMUNICATIONS FORUM

function beyond its appearance. She explained that at one point Intel had tried to obtain a copyright on its semiconductor chips with the argument that they were derivative works of copyrighted drawings of chip design. The current law says that when you make a picture of a "useful" article you do not by getting a copyright on the picture end up with a copyright on the useful article it depicts. Samuelson also went on to discuss several other provisions of copyright law that demonstrate the exclusivity of patent and copyright subject matter. For instance, Section 102B of the U.S. Copyright Laws states that copyright does not protect "processes, procedures, systems and methods of operation," while patent law specifically provides protection for processes.

After providing the audience with an overview of the current copyright and patent rules she admits that these work well when "machines are machines and writings are writings, but there is a dilemma with computer software because it has characteristics of both." In her opinion, software is "too much of a machine to fit comfortably into a copyright system and too much of a writing to fit comfortably into a patent system." She believes that all the confusion we have had to date stems from the fact that the laws do not recognize the existence of "hybrids."

Samuelson has been trying to develop a theory which would explain what it is about software that should be covered by patent and what it is about software that should be covered by copyright. She noted that she doubts anyone else in the country has gotten a handle on this. In response to a recent survey she sent to members of the American Intellectual Property Law Association she noted the variety of responses she received on questions about what things are patentable and which are protectable by copyright. Samuelson emphasized that for the most part lawyers practice either patent law or copyright law which inhibits understanding of the relationship between the two kinds of law.

Samuelson's major concern was that people would try to use copyright law to protect aspects of software that are more properly protectable by patent law. This is disturbing because patent law has a high creativity standard and a short duration, whereas copyright law has a low originality standard and a long duration. She refers to the Crayola standard since you draw something on a piece of paper and think it should be called

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

COMMUNICATIONS FORUM

"original." In a copyright sense there doesn't have to be any creativity since "originality" simply means it owes its originality to its author. Also, copyright protection lasts for the life of the author plus fifty years, a very long a period of time.

Samuelson noted that when you have a patent system that works there is a lot of incrementable improvement information that is available in the public domain, while in the copyright area the standard is less clearly directed to leaving anything in the public domain. She noted that there are some interesting decisions being made by people everyday whether to opt for patent or copyright protection. She told the audience that she was bothered by a recent conversation with an AT&T employee on this subject who preferred to use copyright protection rather than patent because AT&T already had patent cross-licensing agreements and this person wanted to change as much as the market would bear for his software products. She also sees an emerging problem with the strategy of the "dual protectionists" who believe that once a patent expires a copyright would be waiting to kick-in to provide continued protection.

In the area of software, Samuelson notes that source code is one area that copyright covers very clearly. The Patent Office has issued a claim that it will not accept claims that are basically source code listings since it will only deal with a level of abstraction above source code. She continues to search for other things in the software area that will specifically be covered by patents and copyrights. She believes that comparable questions were raised in other industries, including games which are patentable and not copyrightable, and recipes which are patentable, but not copyrightable.

Samuelson's theory is that "copyright protection ought to be 'thin' to protect the code and perhaps the explanatory text on the screen, while patents should should protect inventive functional designs for software, leaving more incremental innovations." She believes that some people might be upset by her recommendations since she believes patents should play an even broader role than they do today.

The last speaker, Dan Bricklin, is currently the President of Software Garden, Inc. He is best known as the creator of

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

COMMUNICATIONS FORUM

VisiCalc, the first electronic spreadsheet program. Bricklin is an MIT and Harvard Business School graduate.

Bricklin focused his talk on how the structure of the pre-packaged software industry might change as a result of broader intellectual property rights. He went on to explain that the typical royalty structure calls for software publishing companies to pay a 10-15% royalty on the gross revenues of the product to software developers. Bricklin emphasized that a sophisticated applications program may involve 10-10,000 patentable processes. He noted that if companies began spending money to obtain software patents for these processes than the current royalty structure would have to change in order for companies to remain profitable.

Bricklin noted that there is an important distinction to be made about most plant or chemical patents, in contrast to software patents. He explained that there is usually one patent that covers the whole product in the case of plant or chemical products. In contrast, a software product can easily involve hundreds of patents for a single product.

Bricklin characterized the software industry as inherently cottage-based. He explained how most of the major advances in the PC industry seem to come out of small shops or out of small development teams. Some examples include WordPerfect Corp., Lotus Corp., and Software Arts. Bricklin noted that with even better tools today one programmer can do even more than he accomplished in the past. He believes that some products should be written by individuals or small groups to achieve better cohesiveness while there is still demand for large companies to handle the larger scale projects. In some cases, Bricklin notes that it is cheaper for a company to go outside and buy a software product rather than develop it themselves. He believes that if the industry had the copyright protection just on the source code it would be cheaper to buy than to make.

Bricklin commented that many people feel software must be "protectable" because it is a product of someone's hard work. In his opinion, "craftmanship" is not protectable and he does not feel that just because you work hard on something, e.g., software, that it should be protectable. He believes we should have



MASSACHUSETTS INSTITUTE OF TECHNOLOGY

COMMUNICATIONS FORUM

patents because patents advance technology, not because patents are inherently good.

Bricklin also cited the problem of "mine fields" in that a software developer often finds out about a related patent after the product has been shipped. He also questions the motives of many lawyers who argue that patents and copyrights be expanded to apply to a broader range of things since it is self-serving. As a developer of software he is also uneasy about patents since he admits having limited knowledge about intellectual property. In reaction to the increase in software patents, Bricklin noted he has been working on lower tech products which involve using information in the public domain.

In conclusion, Bricklin thinks that it is going to take ten years to work things out and that there will be a change in the industry while some of the protection comes in. Bricklin believes that the software industry has "done very well and has moved ahead very quickly because it is satisfying the demands of customers." He believes that given the software industry is working well, there is not room for the expanding role of patents and copyrights to fix anything.