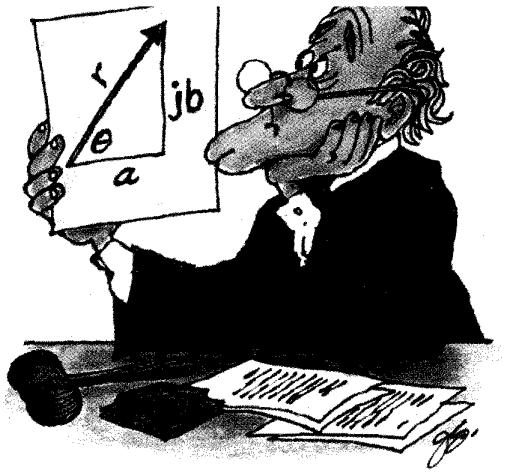
Micro Law



Patenting software, revisited

Richard H. Stern

Ablondi, Foster, Sobin & Davidow

1130 Connecticut Ave., NW, Washington, DC 20036

102503.2466@ compuserve.com

The August 1995 Micro Law reported that the Patent and Trademark Office (PTO) had thrown in the towel in its 30-year battle against patenting algorithms and other software abstractions. In June 1995, the PTO had issued draft guidelines on software patents and asked for public comment. The guidelines indicated that the PTO would be willing to let anybody have a patent on a computer program procedure as long as the patent was limited to the procedure when encoded into a machine-readable floppy disk, DRAM chip, or other storage medium or memory. Since that seemed like no limitation at all, as a practical matter, it appeared that the PTO was declaring open season on patents for algorithms, data structures, and mathematical formulas.

The PTO did express two important reservations that it said it would make to protect the public interest against unduly broad software patents. First, it would allow no patents on music or poetry. The draft guidelines assured us that the PTO would not issue patents on any "known machine-readable storage medium that is encoded with data representing creative or artistic expression (for example, a work of music, art, or literature)."

Second, even in the technological domain, the PTO would exercise restraint by refusing patents on software encoded in a storage medium that was not "capable of causing a computer to operate in a particular manner." (The PTO did not explain what kind of software or storage medium for software does not cause a computer to operate in a particular manner. Write-only memory is one possibility. It is left as an exercise for the reader to determine what the PTO meant.)

Aside from protecting the public against patents on music and on software stored in a nonfunctional storage medium, however, the PTO seemed to be indicating that from now on, anything goes for software patents.

The August 1995 Micro Law suggested that there were at least two things wrong with the PTO's proposal. One was that intermediates in the software chain of distribution (for example, software retailers, such as a reseller of MS-DOS 6 with infringing compression software in it) would become liable for serious damages to software patent owners. This would occur even though they did not know, and had no practical way to learn, that they were dealing in infringing goods. (This problem is akin to that of a net access provider if held liable for copyright infringement because a subscriber uploads infringing material.) The other suggestion was that algorithm patents would cover applications way beyond the teachings of the patent on the algorithm in question (for example, as if Morse's telegraph patent were extended to cover fax machines).

Another possible problem with the proposed software patents is their potentially devastating effect on reverse engineering. Under current copyright law precedent, designers can (as legally permissible fair use) disassemble a computer program to obtain information from it needed to create compatibility with hardware or software to be used in connection with the program. Even though the process involves making "copies" of the original, copyrighted program—in RAM or in nonvolatile storage—the copying is excused as necessary to serve the public interest. (Whether similar reverse-engineering conduct for the purpose of creating a computer program competitive with the original, copyrighted program is permitted, however, is open to question.)

Yet, if the same computer program can in the future be protected by a patent on the underlying procedure when embodied in machine-readable code stored on a floppy disk, any reverse engineering of that code on a floppy disk will

continued on p. 77

0272-1732/96/$5.00 © 1996 IEEE

Micro Law

continued from p. 5

involve patent infringement. A major fly in the reverse-engineering ointment is that patent law does not have a doctrine of fair use comparable to that of copyright law. Making and using something patented (which would occur in reverse engineering), for purposes of developing a competitive product, is ordinarily an act of patent infringement. Hence, software reverse engineering, now at least partially tolerated under the new regime of software patents, would become definitely illegal.

Probably, there are other serious problems with the PTO's proposed software patents, but I haven't figured them out. Perhaps, others contribute to the advancement of thinking on this subject, but so far there has been only negligible comment on the PTO's software patent proposals. Those inclined to enlighten the Commissioner (Bruce Lehman) on the matter can do so at http://www.cos01@uspto.gov.

Revisions

All of this is by way of introducing the information that the PTO still has not issued its guidelines in final form. In fact, the PTO has recently issued for public discussion a new set of draft guidelines that seem to back away from the previous set released last June.

The new draft guidelines are longer than the preceding draft by an order of magnitude. But they are not very much more informative. They are less clear, if anything, than the previous version on whether the PTO is willing to issue a patent on an algorithm or other software idea, as long as it is presented in the form of code on a floppy disk.

The present version is written much more in the form of instructions to the PTO's examining staff for checking software patent applications than it is in the form of telling the public what kind of software patents will now be allowed. Not surprisingly, therefore, it is not easy to discern the answer to that question from the mass of verbiage in the draft guidelines.

In the thousands of words of the new version, the closest thing to an answer is in a passage of warnings to the PTO staff. They must not allow patents on "means" for performing a mathematical procedure when there is not a disclosure in the patent application of "specific structures or materials" to act as the means for performing the procedure. The guidelines say:

> For example, if the applicant discloses only the function to be performed and provides no express, implied or inherent disclosure of hardware or a combination of hardware and software that performs the functions, the application has not disclosed any "structure" which corresponds to the claimed means.

Therefore the PTO examiner should reject the claimed patent protection.

Then, the guidelines explain this further by contrasting with it what a patent application should state and therefore when the examiner should allow claimed patent protection:

> In contrast, if the corresponding structure is disclosed to be a memory or logic circuit that has been configured in some manner to perform that function (for example, using a defined computer program), the application has disclosed "structure."

(Somewhat earlier, the guidelines state that a means for performing a function can be a logic circuit or "a computer memory encoded with executable instructions representing a computer program that can cause a computer to function in a particular fashion." Clearly, in the preceding statement about structure, "memory" is a noun rather than an adjective modifying "circuit," and thus it refers to any kind of storage medium, such as shift register, DRAM, EPROM, or floppy disk.)

What this seems to say, behind all the mystical incantations about "structure," is that if the patent application discloses a defined computer program as stored on a floppy disk or in some other memory device, PTO examiners should allow a patent. If this interpretation is right, we are right back to where we were almost a year ago when the PTO issued the first version of the guidelines. The PTO will grant a patent in the following form:

> A floppy disk or other storage medium containing machine-readable computer code for carrying out the following procedure in a general-purpose digital computer: (now, recite a series of steps for doing mathematical operations such as summing a first and second parameter, integrating the sum with respect to time, and so on).

Since the patent is on a floppy disk (that is, encoded storage medium), the floppy disk will not be limited in its use, for purposes of determining patent infringement, to any particular field of technology. The patent will cover the procedure, in whatever applications it has, as long as the application requires use of a computer to execute the class of procedure encoded on the floppy disk. This must be contrasted with an ordinary patent, which is limited to particular apparatus or particular physical transformations. To claim in an ordinary patent the same kind of procedure as that for which the floppy has code, as above, you would have to mention that the procedure was being carried out in a rubber-molding machine, microwave oven, robotics materials handler, or milling machine, or that it operated on molded rubber articles, raw foods, pieces of metal coming down an assembly line, and so on. You could not claim the procedure divorced from all context, but apparently you can in floppy-disk patents.

What are the consequences?

What does it mean to give out patents on a floppy disk encoded with machine-readable code for performing a function or procedure? Why should we care about that? Does it have any more than academic consequences? Consider those questions in light of the following hypothetical case.

I am the first to discover a procedure for measuring the operation of spring and dashpot systems. (A system of that type is illustrated in Figure 1.) It so happens that many people are interested in spring-dashpot technology, for it is useful for operating mechanical back scrubbers, back scratchers, and

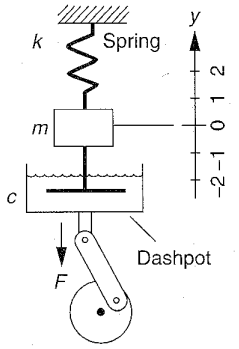


Figure 1. Mechanical back scratcher system.

eraser carriers. Each of them operates on the principle of a reciprocating stroke in which you desire to control, by appropriate means, the linear excursion and steady-state speed of the back scrubber, back scratcher, or eraser. I have discovered that the system of Figure 1 can be modeled by the following equations:

$$F(t) = F_0 \sin \omega t$$

$$= m\frac{d^2y}{dt^2} + c\frac{dy}{dt} + ky$$

$$F(\omega) = y\left[c + j\left(\omega m - \frac{k}{\omega}\right)\right] = yZ$$

$$y = \frac{F}{Z}$$

Here, k is the spring constant, c is the dashpot constant, m is mass, F is force (sum of moving force – friction), and y is the displacement.

Say that I want a steady-state back-scratching displacement of, say, 0.25 meters, and I plan to have a stroke/second rate of 0.314, which results in an ω of 0.1. Then, it is a relatively simple matter to determine the appropriate parameter values.

I realize that this invention will be of great commercial value in controlling automatic mechanical back scratchers and similar reciprocating devices. So, I take advantage of the patent system to protect my intellectual creativity. I want a patent on methods and systems for controlling displacement of reciprocating devices in accordance with a predetermined speed and the like, in which you select appropriate parameters in accordance with these equations. I therefore explain all of this in a patent application and claim the methods and systems.

In addition, I certainly want protection for the procedures in the form of "a computer memory encoded with executable instructions representing a computer program that can cause a computer to function in a particular fashion." Here the "particular fashion" is working out the equation values just indicated. I therefore claim a computer memory (including, without limitation, a floppy or hard disk) coded with machine-readable instructions to solve the equations for whatever parameter values I need to get a solution for F, y, ω, or whatever. Under the guidelines, the PTO would presumably have to allow the claim.

Assume now that a year after my patent issues, you become the first to discover that applying an electromagnetic force (EMF) to a series connection (or other network) of a resistor, inductor, and capacitor can be rationalized in a way that permits control of current through an electrical device. You discover that EMF E can be represented as a function of current i, resistance R, and so forth. Among other things, you develop the following learning on the subject, which can be represented by these equations:

$$E(t) = iR + \frac{1}{C}\int i\,dt + L\frac{di}{dt} = \sin \omega t$$

$$\frac{dE}{dt} = R\frac{di}{dt} + \frac{1}{C}i + L\frac{d^2i}{dt^2} = \cos \omega t$$

$$E(\omega) = i\left[R + j\left(\omega L - \frac{1}{\omega C}\right)\right] = iZ$$

$$i = \frac{E}{Z}$$

You decide to commercialize your discoveries, by manufacturing and selling computerized devices that will control current i through an electrical device by means of manipulations of appropriate parameters in accordance with the learning embodied in the foregoing equations. You will have a slight problem.

My patent covers your procedure.

Implications

All of this is an elaboration on the problem caused to society if Morse's telegraph patent is allowed to cover the fax machine. The PTO's software guidelines seem to allow someone who comes up with a procedure that can be expressed algorithmically, or otherwise in a manner lending itself to computerization, to dominate the procedure, as such. The guidelines do not limit a floppy-disk patent on a procedure to the original application, but rather allow the patent also to cover other unknown or even unforeseeable applications that others later discover. If a later system operates in accordance with the same differential equations, or algorithmic system, as a previously discovered and patented one does, the patent covers the later system.

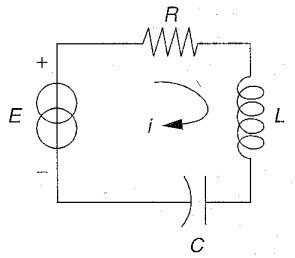


Figure 2. RLC circuit.

That appears to be very unwise and also more potentially damaging than beneficial to software progress. The grant of a patent to a patent owner is the result of a bargain between the public and the patentee. The patentee instructs the public in how to practice a new invention; in return, the public gives the patentee approximately two decades of exclusive right (a monopoly) to practice the invention that he taught the public. That bargain is not kept when the patentee teaches the public one thing, and then acquires a patent on that and also something else he never taught. Not only is the public not getting its fair consideration for the patent, but the effect of granting an excessive patent creates a disincentive to others to develop and commercialize inventions in the excess monopoly area.

If Morse teaches the public how to make and use telegraphs, and then gets a patent on both telegraphs (which he taught) and fax machines

(which he did not teach), why should you then invest your sweat equity into inventing the fax machine? Morse already holds a patent on fax machines. And why should a source of capital invest money into your project, if you are so unwise as to persist?

Or take my hypothetical case. I devised a way to control mechanical back scratchers and similar devices. I got a floppy-disk software patent on the software system for doing this. That is a patent on a floppy disk in which is encoded the software method of carrying out a mathematical procedure in a computer corresponding to the relevant differential equations.

It later turns out that the same procedure can be applied in another field of technology. It is not uncommon for the same second-order differential equation to apply to several systems. For example, surely many other fields than RLC circuits and spring-dashpot systems operate in accordance with an equation of the form $F(x) = ay'' + by' + cy$. Then a floppy disk can contain instructions for reaching the solution represented by $F = x(R + jI)$, for any of them.

It is not in the public interest that my floppy-disk patent, which has not taught the public about RLC circuits, should nonetheless dominate computerized use of the subsequently developed field. It is neither fair nor rational to give me a monopoly that I did not earn.