

Adobe - Software Patent Policy

*Public Hearing on Use of the Patent System to Protect Software Related Inventions
Transcript of Proceedings Wednesday, January 26, 1994 San Jose Convention Center*

Douglas Brotz, Principal Scientist, Adobe Systems, Inc.

Mr. Brotz: Good morning, Mr. Secretary and members of the Panel. My name is Douglas Brotz. I'm Principal Scientist at Adobe Systems, Incorporated, and I am representing the views of Adobe Systems as well as my own. Adobe is a software company based in Mountain View, California. We are most well-known for our PostScript language and interpreter which provides foundation for desktop and electronic publishing.

Although I am a computer scientist, I became involved in patents when Adobe was contacted by another company regarding Adobe's possible infringement of a patent. I'm currently Adobe's technical advisor to our patent attorneys.

Let me make my position on the patentability of software clear. I believe that software per se should not be allowed patent protection. I take this position as the creator of software and as the beneficiary of the rewards that innovative software can bring in the marketplace. I do not take this position because I or my company are eager to steal the ideas of others in our industry. Adobe has built its business by creating new markets with new software. We take this position because it is the best policy for maintaining a healthy software industry, where innovation can prosper.

The problems inherent in certain aspects of the patent process for software-related inventions are well-known, the difficulties of finding and citing prior art, the problems of obviousness, the difficulties of adequate specifications for software are a few of those problems. However, I argue that software should not be patented, not because it is difficult to do so, but because it is wrong to do so.

The software marketplace requires constant innovation regardless of whether the computer programs can be patented or not. Indeed, the fundamental computer programs and concepts on which the entire industry is based were conceived in an era when software was considered to be unpatentable.

For example, when we at Adobe founded a company on the concept of software to revolutionize the world of printing, we believed that there was no possibility of patenting our work. That belief did not stop us from creating that software, nor did it deter the savvy venture capitalists who helped us with the early investment. We have done very well despite our having no patents on our original work.

On the other hand, the emergence in recent years of patents on software has hurt Adobe and the industry. A "patent litigation tax" is one impediment to our financial health that our industry can ill-afford. Resources that could have been used to further innovation have been diverted to the patent problem. Engineers and scientists such as myself who could have been creating new software instead are working on analyzing patents, applying for patents and preparing defenses. Revenues are being sunk into legal costs instead of into research and development. It is clear to me that the Constitutional mandate to promote progress in the useful arts is not served by the issuance of patents on software.

Let me illustrate this burden with some figures. The case *Information International Incorporated v. Adobe, et al.*, was filed five years ago. Last year the trial court ruled for Adobe, finding no infringement. In December the Appeals Court for the Federal Circuit unanimously affirmed that judgment. Yet, in that time, it has cost Adobe over four and a half million dollars in legal fees and expenses. I myself have spent over three thousand five hundred hours of my time -- that's equivalent

to almost two years of working time -- and at least another thousand hours was spent by others at Adobe. The Chairman of the Board spent a month at the trial. This type of company behavior would not be high on anyone's list of ways to promote progress.

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.

There does exist a perfectly adequate vehicle to protect creator's rights in this industry, the Copyright Law. The nature of software is that it is a writing, an expression of mathematical ideas. The copyright law protects this expression, and it does so without requiring costly and time-consuming proceedings. For people working in the fast-paced software industry, the way a copyright is created is idea. While feverishly working to meet deadlines, there is no need to explain what you've done to a government agency. The very act of writing the software confers the copyright on it.

Furthermore, the copyright law confers the correct level of protection on computer software. Regardless of what current regulations may say, the fact is that all computer programs express mathematical algorithms. Every part of every computer program manipulates numbers with logic. Any software that performs any task does so through mathematics. It is inconsistent to hold that mathematic algorithms are unpatentable while granting patents on systems composed of software.

If the Patent Office were truly following the law it would recognize the inherent mathematical nature of software and it would not grant patents to software-based inventions. In the last decade the Patent Office has been granting patents on software and algorithms regardless of superficial attempts to cast claims as systems methods or processes. The Supreme Court did not say in *Diamond v. Diehr* that pure software inventions are patentable. By adopting this position in its recent practice, the Patent Office has made a dangerous step that could decimate the very industry it wishes to protect.

Whenever the Patent Office grants a software patent, it grants a right to the patent-holder to devastate innocent businesses. Due to the arcane nature of this technology, our courts find it very difficult to distinguish frivolous software patent lawsuits from legitimate ones. As a result, a frivolous plaintiff is in a very strong blackmailing position, where a defendant can look forward either to an extortionate settlement or enormous legal costs. An excellent remedy would be to change our law to allow a successful defendant to recoup legal costs in patent cases. Until that day arrives, at least our Patent Office can refrain from granting these dubious patents.

We have heard today from proponents of software patents who will claim that these patents can protect the independent inventor. This belief is a delusion. The expensive patent process protects large, methodical corporations that can afford to apply for scores of patents much more than it protects the poorly-capitalized lone inventor, and when that inventor tries to produce his invention he may well find that those large corporations can ruin his own business with their large software patent portfolios.

In summary, these are my main points:

- the software industry thrived without patents, creating its fundamental base in an era of no software patents;
- software patents harm the industry, with no corresponding benefit;
- software embodies mathematical algorithms;
- the law, starting with the Constitution, argues against patents for software-related inventions;
- and last, the proper form of protection for software is copyright.

As a postscript to the figures on the patent lawsuit that I discussed before, the final figure is actually not in. Although Adobe has been successful twice already, the plaintiffs are asking for reconsideration of the unanimous appeal judgment against them. These kinds of festering sores are what our country can ill-afford when we are trying to lead the world in creative industry.

Thank you.

