From: J.B. Nicholson-Owens [e-mail redacted]
Sent: Sunday, September 26, 2010 8:28 PM
To: Bilski_Guidance
Cc: [e-mail redacted]
Subject: Bilski guidance

I'm writing to tell you some of my thoughts on software patents and to encourage the USPTO to take a strong stance against software patents (patents on ideas used for developing computer software).

The public is under the threat of losing software patent lawsuits merely by using computer software. The public is largely unaware of this threat and taking a strong stance against software patents will help eliminate this threat.

Around the time the Apple computer company sold Macs with the Hypercard program, a man named Paul Heckel held a patent which he and his lawyer believed Apple had implemented in Hypercard without Heckel's permission.
  Heckel informed Apple he and his lawyer believe Apple was implementing his patented ideas and Heckel apparently expected to negotiate payment with Apple. Apple refused to pay Heckel. So Heckel told Apple that he'd sue Apple's customers for patent infringement instead, since they were also using his patented ideas without permission. Apple took notice of this, probably because they didn't want to be known as a company that leads their users to losing patent lawsuits, and negotiated payment with Heckel. I think its reasonable to assume that Apple's lawyers considered Heckel's claim carefully. I don't think Heckel would have gotten a penny from Apple if he had no power. I think Heckel's ability to get money from Apple shows the threat the public is under from patent holders. Removing software patents entirely would eliminate this threat and let computer users develop and use software without placing the public at risk in this way.

The public's reliance on computers means that just about everyone will be subject to software patents (not just software developers). The public is bound to increasingly identify computers of all kinds as devices they own and ought to control like any other object they own.
Software patents would interfere with that control by subjecting the general public to possibly losing a patent lawsuit if the public improves or modifies a program without licensing the relevant patent(s) first (which the public is highly unlikely to do and probably can't afford to do). Hiring a programmer to help relieves none of the pressure; the triple-damages punishment for knowingly infringing a patent discourages programmers from reading patents. I don't think it's fair, ethical, or good public policy to restrict the general public based on which ideas they are allowed to use.

When personal computers started becoming popular in the 1980's software patents either didn't exist or weren't used. But we enjoyed a wide array of software to do all sorts of tasks, including software developed by businesses.

Business don't need a monopoly to develop the ideas one uses in computer software nor implement those ideas in code, some businesses merely want that power. Commercial software development can compete on price and features, it should not compete by locking out other computer users from implementing ideas (as patents are designed to do). We should encourage businesses to return to competing based on developing better software that meets people's needs, not developing more effective strategies for locking users into one implementation of an idea or preventing competition by prohibiting alternative expressions of ideas. I understand that today's businesses obtain software patents just to deal with the hassle of other businesses' software patents!

I write computer software and I am adversely affected by software patents. I license my programs to others under terms that allow anyone who gets a copy of the program to share and modify the program. But software patents prohibit me from implementing certain algorithms. For example, some of the software I write creates, manipulates, and plays audio files. I cannot distribute software to create or play MP3 files due to the patents covering algorithms to encode and decode MP3 files; I cannot distribute such software without subjecting myself and my users to the risk of losing a patent lawsuit. As a result my software is needlessly less compatible (and thus less competitive) than I'd like it to be. The US should encourage competition by letting us all develop software that does what we wish.

The adverse impact on the public, the sheer lack of necessity, and my own experience are some of the reasons I think that the USPTO should take a strong stance against software patents and the USPTO should no longer issue software patents. I think such a stance would be in line with the recent Bilski SCOTUS decision because the Supreme Court wants a more restricted guideline for patent eligibility than was believed to be the case before Bilski was decided.