From: David A. Wheeler [e-mail redacted]
Sent: Friday, September 24, 2010 3:21 PM
To: Bilski_Guidance
Cc: [e-mail redacted]
Subject: Stop permitting software patents

The new patent office guidance needs to *stop* permitting software patents, to comply with Bilski and previous Supreme Court rulings.

The U.S. Supreme Court has *never* ruled that software should be patented. In Gottschalk v. Benson (1972), the United States Supreme Court ruled that a patent for a process should not be allowed if its "practical effect would be a patent on the algorithm itself". The more recent Bilski v. Kappos makes the boundaries of patent eligibility even more narrow, and the PTO's rules must thus be narrowed as well.

Fundamentally, software consists only of mathematics, which is not patentable. The combination of software with a general-purpose computer is obvious, so implementing software on a general-purpose computer is also not patentable. If a computer-related claim does not require any special attached peripherals (e.g., attachments for rubber processing), then it is merely an abstract algorithm and should be barred on its face.

Indeed, I believe software patents are unconstitutional, because they have no constitutional justification. The Constitution says that the sole purpose of patents is to "promote the progress of science and useful arts". Yet numerous studies and reports have shown that software patents retard progress, with a great cost to the economy. Without Constitutional justification for them, they must be eliminated. My own work identifying the major innovations in software (http://www.dwheeler.com/innovation/innovation.html) is highly-ranked by Google when searching for "software innovations". After analyzing *real* innovation in software, as well as reading the many studies debunking the myth that software patents were helpful to innovation, I reported that "Software patents are actually harmful, not helpful, to software innovation, as confirmed by a myriad of data." There is practically no evidence that patents have been helpful to innovation in software, and vast evidence that it is harmful.

Software is *ALREADY* covered by copyright law. We do not let authors (or their employers) patent plots of the books they write. For the same reason, there is no need to let programmers (or their employers) patent algorithms of the software they write. This creep of patenting into the domain covered by copyright is harmful to society, and is a creep not justified by law.

Software patents are clogging the court system, and are causing the entire patent system to fall into disrepute among many. The recent paper "Patent Quality and Settlement among Repeat Patent Litigants" (Allison et al) examined "repeat patent plaintiffs" (those who sue eight or more times on the same patents). They thought such "patents should be among the strongest, according to all economic measures of patent quality... But, to our surprise, we find that when they do go to trial or judgment, overwhelmingly they lose". They determined a primary cause was the presence of software patents and patent trolls (aka non-practicing entities or NPEs); weak patents were often from those sources. They concluded that "If software and NPE [patent troll] patents are overwhelmingly bad - either invalid or overclaimed - the social benefit of allowing them may well be OUTWEIGHED BY THE HARM THEY CAUSE [emphasis mine] ... The patents and patentees that occupy the most time and attention in court and in public policy debates... are astonishingly weak. Non-practicing entities [NPEs, aka patent trolls] and software patentees almost never win their cases. That may be a good thing, if you believe that most software patents are bad or that NPEs are bad for society. But it certainly means that the patent system is wasting more of its time than expected dealing with weak patents."

Permitting software patents is entirely a PTO invention. Historically, during the time when software innovations were particularly common, software patents were NOT allowed, and there was never any evidence that allowing software patents solved a problem. No law or Supreme Court ruling has specifically said they were permitted. Recent mis-interpretations by the PTO have allowed them; since it is the PTO that made that determination, the PTO can re-examine these rules and fix this mis-interpretation.

This involves more than money; it involves freedom. At one time computers were exotic and rarely used. Today, computers are everywhere; they essentially control everything, and human communication depends on them. Our government was founded to "ensure the blessings of liberty, to ourselves and our posterity". Yet this liberty - aka freedom - now fundamentally depends on the ability of people to control the computers around them. When people can control the computers around them, they can control the environment around them. Software patents instead create a legal barrier that ensures that people will stay controlled by their computers instead, because people are no longer allowed to modify their computers to implement many useful tasks.

More information about why software should no longer be permitted are here:
http://www.dwheeler.com/essays/software-patents.html

Please note, I am a U.S. citizen.

Thank you for your time!

--- David A. Wheeler