From: Ken Settle [e-mail redacted]
Sent: Sunday, September 26, 2010 3:42 PM
To: Bilski_Guidance
Subject: Please stop accepting software patents!

Hello,

I am a long-time software developer, who remembers a time before software patents were prevalent. Now, they are an epidemic! As a software developer over the last few decades, I have no doubt innocently infringed on thousands of software patents by merely doing my job. This is certainly the case for any software developer who has been working for any period of time. Software developers can't even look at software patents that are relevant to their work, because if they know about them, they risk getting sued for treble damages.

Software should be thought of as a very detailed contract between the software developer and the computer, written in a specific language that the computer can understand. Try to imagine the legal ramifications of allowing patents on specific types of legal contracts! That would create the kind of problems for lawyers that software developers are facing. They could be sued for coming up something obvious, like an employment contract, because some other lawyer had a patent on it.

Software patents should be excluded on legal grounds, since all software is math. Math or "algorithms" were not intended to be patentable, and they were specifically disallowed by the Supreme Court in their Bilski v. Kappos decision. The combination of such software with a general-purpose computer is obvious. Any arguments that software physically "changes" a general-purpose computer are inaccurate at best. The mere storage of computer programs or data on a general-purpose computer hard disk does not represent a meaningful physical change in matter (given that was your previous test, it has not been met). The change of magnetic states on a hard disk do not relate in any way to the function of the software or any other data that is stored there. That's like patenting a series of bits of digital data that makes up part of a digital file that represents a movie, television show, or music. Also, more devices should be considered to be "general-purpose computers," including smartphones, tablets, cameras, video game systems, MP3 players, televisions, wireless routers, and nearly anything else that has a CPU in it. You should disallow software patents that are "narrowed" to only apply to these devices. They are all general-purpose computers, disguised as dedicated devices. Deep down, they are no different than your desktop computer or laptop.

Patents are supposed to promote the creation of new and useful inventions. When Microsoft started out, software patents weren't prevalent. Microsoft didn't need to worry about innocently infringing on someone's dubious patents for any idea that they came up with in the normal course of software development, and getting sued for it. Now Microsoft patents every possible idea that they have, even things that they have no use for, just to prevent someone else from doing them, or for cross-licensing ammo in a game of Mutually Assured Patent Destruction cross-licensing with other giant corporations. Yet they still get sued by many third parties that buy up software patents and sue everyone, but create nothing of value themselves! Large companies and small companies would both benefit from software patents being abolished. If anything, innovation has been slowed down by the legal minefield that software patents represent.

Software development consists of continuous "invention" of simple ideas that are "re-invented" by everyone who attempts to solve the same problem. The non-obviousness requirement of patents is frequently violated by software patents. If you were to ask ten software developers to solve a given problem, the majority of them (if not all of them) would come up with a solution that was similar or identical to the prospective software patent, even if none of them had any prior knowledge of the software patent application. Also, the disclosure of a "best method" for implementing software patents is often missing. The "best method" would be the software's source code, which is rarely given. As a result, the software patents often don't describe in specificity a manner of practicing the invention, making the software patent effectively useless to the public domain once it has expired. Frequently software patents are over-broad, attempting to control any possible implementation of an idea. Patents aren't supposed to cover ideas, only inventions. Also, because the pace of innovation in software is so rapid, by the time a software patent term has expired, the public domain receives a useless "invention" because it is almost always obsolete. Contrast this with a patented mechanical device which frequently has a useful life that is much longer than the term of the patent. That is what patents were originally designed to protect and promote!

According to press reports, there is a huge backlog (several hundred thousand) of software patents alone. This is creating a situation where the USPTO is unable to keep up with patent applications, and the delays for patent acceptance are becoming unacceptable. If you stop accepting software patents from being filed or examined, you will improve the patent office's efficiency immensely.

Prior art is difficult to find for software patents, which makes the patent examiner's job even more difficult. The lack of visibility into prior art results in multitudes of software patents being granted for "inventions" that were previously made by someone else, and often practiced and sold to the public (or given away for free) for years before these software patents were even conceived of. The original inventor is often sued by the recipient of these dubious patents, and they must either pay exorbitant amounts of money to avoid a lawsuit, or possibly more money defending their invention in court, while the dubious patent holder gets to slander their good name in the press for years! This is a very egregious form of injustice that is perpetrated by allowing software patents.

In closing, software can and should be protected by copyrights. It is a writing. It is not an invention. Abolishing software patents will certainly promote innovation to a much greater extent than continuing to accept them.

Sincerely,

Ken Settle