From: Gordon Cooke [e-mail redacted]
Sent: Monday, September 27, 2010 11:33 PM
To: Bilski_Guidance
Subject: Patent Guidance

Sir/Ma'am:

Ive learned that the USPTO is currently reviewing its guidance about patent-ability following the Bilski decision. As a mechanical engineer Im familiar with the need for protection of IP, I also work with and create software on a regular basis. I would strongly suggest that any new guidance should eliminate, or at least strongly restrict, software patents.

All software is nothing more than mathematical algorithm. Even basic introductory courses in computer science point out that software is an algorithm. It is a completely discrete set of instructions that when followed produced repeatable results. This will remain the case until someone invents a new kid of processor (such an invention would be a hardware improvement and not a software one).

Software is nothing more than a set of instructions for the real machine (the hardware of the computer). Allowing software then would mena that the patent office should also be granting patents for all sets of instructions for machines. If this is the case then I would like to file for patents the thread layout of plaid cloth. Someone else has patented the loom. But I want to load the weft with 3 red threads, 3 green threads and two white threads (repeat) and use the same pattern on the warp. Oh and another patent for 4 red followed by 4 blue..... Or maybe I can file for a patent that just says I load any alternating/repeating pattern and use the same pattern on the both warp and weft so I can pattent all plaids, even ones I havent thought of-- on someone elses existing loom whose patent already says load with thread and create cloth. Software instructions are no different. The basic instruction set to drie the hardware was created by the original hardware inventor- all software then uses that same instruction set to use the hardware.

An additional issue with the current software pattents is the overbroad claims of them. Patents are supposed to have enough information in them that someone skilled in the art could recreate the work. But many software patents make claims about what the software does, without describing how. In order to adequately describe HOW a software 'invention' works would be to fully describe the algorithm with DETAILED flowcharts, or to provide the source code (at least in psuedo code). A patent that does not includes this would have one of two problems. If someone skiled in the art of programming can read what the software does, without any cde or detaied algorithm, then it must be obvious, and therefore not novel and not patentable. If however it i s not obvious how one would accomplish the claims, then withoutthe code or algorithm the 'invention' is not fully described and should not be awarded a patent. Alowing patents for claims without a fully described method allows patent holders not only a monopoly on their intelectually created work, but also allows them to claim infringement for novel (possible better) solutions they hadnt even thought of. This stifles inovation.

The purpose of patents is supposed to be to encourage innovation. An inventor is allowed a limited time monopoly on their idea in exchange for divulging publicly how it works so that others can build off of it. This encourages competition & innovation (in a few years anyone can make my product so I better work on a newer version & ohh look what he did I see an improvement to it.) The alternative is that the inventor keeps theri invention proprietary as a trade secret. They are free to do this for as long as hey can keep the secret, however, at any time a competitor could discover the same idea and use it freely. Patents offer a limited protection (even if discovered independently a competitor cant use it if I discover first) but force disclosure. Somehow the software industry has managed to make an end run around this entire purpose. They have been allowed patents to maintain a monopoly, even to stop others make independent discoveries, and yet have been allowed to keep proprietary trade secrets on how those inventions are achieved. With all the patents in Microsoft Windows I should be able to pull the information from he patent office and build an entire replica of MS Windows and MS Office that I simply put together and sell the day after the patent expires, and it should work fine without any R&D on my part. But this is not possible. by anaogy, it is the software equivalent of me claiming I have a device that you talk to, ask it for any food, and it will assemble the exact food you requested , assembling it out of some store of organic precursors inside. I dont describe how it works or how it assembles the food, but please give me a patents- no one else has doent it so it is novel. maybe I show you a working prototype but the examiner cant look inside to see how the molecules are manipulated. Perhaps I use lasers and just dont write that in the application. Then someone else comes along and invents a beter version- they use soem kind of magnetic field manipulation that uses less power and is safer. I didnt think of that. But my patent just says 'makes food' So I use my patent to stop them from distributing their better product that works in a way I didnt think of. In software this seems to not only be allowed, but common.

Patent applications are supposed to make the invention inspect-able by the patent examiner, including providing a working prototype. This is to allow the examiner to see and verify exactly how the invention works and that it is novel. The only way software could be provided to an an examiner in a way that meets this need is if it includes the source code. Binary only versions may demonstrate the output of the software but prevents an examiner from judging the novelty or obviousness of the solution. Therefore, if allowed, software patents would require source code- and as the entire applications is part of public record, the source code must be made open.

Software patents do nothing to generate progress in the sciences or useful arts.

20 years ago software was commonly covered by copyright protections and not patents. The software industry still routinely uses copyright notices on software. Yet patents are allowed for the very same software. By definition are not these two systems mutually exclusive? How can an entire industry be allowed to use both protections for the very same product?

I strongly encourage you to end software patents, or please at list limit them and require

full descriptions of the 'invention' in the patent (ie source code)
Computer hardware (procesors, memory, video cards...) are clearly patent eligible inventions. Some software drivers that are tightly coupled with specific hardware (ie will only run on that single chip and are part of the hardwares function- or the hardware wot function without it) may be eligible also. But general purpose software, with claims that they apply to any computer (MS Office will run on Intell 32bit chips, AMD x64 chips, and even Apple computers) should not be allowed patents (copyright is much more appropriate)

Sincerely
Gordon Cooke