From: Dominic Mazzoni [e-mail redacted]
Sent: Monday, September 27, 2010 10:09 AM
To: Bilski_Guidance
Cc: [e-mail redacted]
Subject: Commentary on software patents

Dear USPTO,

I have been a software engineering professional for 10 years and I have been writing software since before I was in college. While I currently work for Google, Inc., these views are entirely my own and not the views of my employer. I am not a lawyer but I am familiar with patent law and I am writing this letter in response to the call for commentary on software patents in the context of Bilski v. Kappos. I would like to encourage the USPTO to end the practice of allowing patents on software algorithms.

It has been my experience that the vast majority of software patents are holding the industry back, stifling innovation, rather than performing the intended purpose of a patent, which is to (1) give an inventor a short-term monopoly on their invention to recoup their development costs, and (2) give back to the general public by releasing all of the details of the patent after its expiration.

The first problem with software patents is how many of them fail the novelty / originality test. Writing computer algorithms is a very specialized field, but nevertheless there are millions worldwide who are skilled in this art. When presented with a particular problem, typically anyone capable in this field should be able to propose a reasonable algorithm to solve the problem. Occasionally some problems are more difficult, where a particularly efficient algorithm eludes all but the best minds, but far more often, the solution is straightforward and obvious to anyone presented with the problem with adequate experience in the field. Unfortunately just about any "obvious" solution seems to be able to get a patent awarded. In many cases there is obvious prior art for these patents, but in other cases there's no prior art because the problem is unique, but that doesn't make the solution any more worthy of patent protection if the solution would be obvious to any competent professional presented with the same problem. There are thousands and thousands of software patents that contribute nothing to the advancement of the field but only serve to create a legal headache for everyone who just wants to do their job.

A second concern is whether even the most novel and original software algorithms should deserve patent protection at all. I believe that they should not, because algorithms are mathematical formulas, and because they are not necessary to provide intellectual property protection to inventors. As an algorithm in the abstract is indistinguishable from a mathematical idea, it should not belong to the person who first discovers the concept, but to all of humankind. The concrete application of this mathematical concept to a machine (a computer) is thus not a novel invention, but simply a creative work. As a creative work, a particular computer program deserves copyright protection, which is quite sufficient to provide income to the author and prevent others from directly profiting from his labor. When the algorithm itself is also patented, though, it often has the effect of eliminating competition not just from building the same device, but even from coming up with a completely unique independent solution to the same problem, because the mathematics behind the algorithm are a fundamental property of nature that cannot be circumvented by creativity. If the problem is 2x+3 = 19, no amount of creativity will produce a solution other than x = 8. Similarly, there are many mathematical problems in nature where there is only one optimal algorithm to solve a problem, and no amount of creativity can bend the laws of nature. As such, the laws of nature themselves should not be considered the invention and property of the first to discover them, even though any concrete implementation of the solution (a computer program) deserves copyright protection.

Thank you for considering this viewpoint.

- Dominic Mazzoni