From: Mike Clark [e-mail redacted]
Sent: Sunday, September 26, 2010 9:14 PM
To: Bilski_Guidance
Cc: [e-mail redacted]
Subject: Bilski v. Kappos

Dear USPTO:

I wanted to chime in on the issue of software patents. I am an independent software vendor, and as such I frequently create original sotware programs using various computer languages, and I use mathematical and procedural algorithms to do so, as implemented in the computer language I am using for the project. This process is very much like writing literature of one sort or another, and in fact literature is not patentable (I am not a lawyer, but that is my understanding). If it were possible to patent natural language constructs, such as verbs, adverbs, adjectives and nouns (quite apart from trademarks and the like, which are covered independently), or even sentences, it would be possible eventually for certain individuals to demand licensing fees for people simply having conversations. This is clearly absurd.

But by the same token, the patenting of software constructs is likewise absurd, simply because by virtue of being built with particles of computing (which is at its base, simple mathematics), the general public -- the vast majority of which are at least capable of writing software, and many do -- would gradually find it more and more difficult to so much as merely use their computers at all, because they could never be sure when they might inadvertantly be violating some patent-holders "rights", and thus be subject to some kind of licensing requirement, or be subject to even harsher legal sanction. The absurdity of such should be clear!

I am unalterably opposed to software patents. If those who build software need protection for their work -- as surely they do -- there exists a completely capable body of regulation, called "copyright law", which they can use. It is not heavy handed, and allows protection without crippling others' rights to use their computers.

I would like to join with others, in urging you to get a grip on the patentability of software. Since I am not a brilliant legal mind, I shall not write further on the subject, but will simply endorse what some others have said on this subject. I subscribe to the sentiments as expressed in the following two paragraphs:

"Software patents hurt individuals by taking away our ability to control the devices that now exert such strong influence on our personal freedoms, including how we interact with each other. Now that computers are near-ubiquitous, it's easier than ever for an individual to create or modify software to perform the specific tasks they want done -- and more important than ever that they be able to do so. But a single software patent can put up an insurmountable, and unjustifiable, legal hurdle for many would-be developers.

"The Supreme Court of the United States has never ruled in favor of the patentability of software. Their decision in *Bilski v. Kappos* further demonstrates that they expect the boundaries of patent eligibility to be drawn more narrowly than they commonly were at the case's outset. The primary point of the decision is that the machine-or-transformation test should not be the sole test for drawing those boundaries. The USPTO can, and should, exclude software from patent eligibility on other legal grounds: because software consists only of mathematics, which is not patentable, and the combination of such software with a general-purpose computer is obvious."

Yours for the reform of the currently ridiculous patentability of the computing equivalent of letters, words, and phrases.

Michael L. Clark
1008 Prospect Ave NE
Olympia, WA 98506
360-754-6761