

From: Stephen Jacob Sekula [e-mail redacted]
Sent: Saturday, September 25, 2010 12:04 PM
To: Bilski_Guidance
Subject: Letter in support of excluding software patents from consideration

Dear Sir or Madam,

Please find attached a formal copy of my letter in support of the exclusion of software patents from consideration by the USPTO. Below is the text of my letter. If you have any questions, please do not hesitate to contact me.

Sincerely,
Stephen Jacob Sekula

=====

I am writing to express my opinion that, in your coming review of USPTO guidance for patent eligibility, submission of software patents be expressly forbidden from consideration by your guidelines.

I am a software writer and a user. Both in my personal and professional lives, software plays a fundamental role. I rely on open-source software to power all of the computers in my home: laptops, desktops, and even my own home-built TV entertainment system. I rely on open-source software to advance my creativity as a physics professor, both in teaching and research. If software were encumbered by patents, I would be unable to innovate in my home, or develop software for my friends and colleagues and freely distribute it, without fear that somebody else has patented the idea of the code and can sue me for breach of patent.

I view this issue - software patents - in the same way that I view poetry. Thousands of poets have, time and time again, written about the same subjects - love, money, death, birth, nature, etc. Poets use our human languages to construct literary solutions to the puzzles of human experience, even though there are a finite number of human experiences. Poets are not allowed to patent the idea of writing such a poem; they may copyright their specific construct, but not patent the idea of the construct.

In the software world, thousands of people have had to solve the same problem. They have each brought their unique software writing talent, using the language of the computer (mathematics), to solve those problems. The reason I choose free software, or choose to write software, is because I don't always agree with (or need) the solution written by a for-profit organization. I might choose to write my own program to do the job, and then distribute it to friends or colleagues who require the same solution. Software patents threaten my creativity by bringing the possibility of a lawsuit, all for my use of mathematical machine language to solve a problem. That would be like patenting a poem about love, and then having the poet sue me if I choose to write and distribute my own love poem. It stifles creativity and creates a legal environment that threatens innovation in the United States.

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.