

From: Dustin DeWeese [e-mail redacted]
Sent: Monday, September 27, 2010 10:16 PM
To: Bilski_Guidance
Cc: [e-mail redacted]
Subject: Software is made of abstract ideas

Software in itself is a set of abstract ideas. The versatility of any piece of well written code bears witness to this. A single algorithm, such as a Kalman filter (http://en.wikipedia.org/wiki/Kalman_filter), or a data structure, such as a Bloom filter (http://en.wikipedia.org/wiki/Bloom_filter), has no meaning by itself.

I am a software developer, and my profession is the combination and modification of these ideas. I would be famous if I invented just one of these foundational ideas. Not only are algorithms and techniques such as these in danger of being patented, but even obvious things, such as Amazon's one-click patent.

As a lone software developer, I cannot afford to do patent searches to find out if code I wrote myself, based only on my experience and formal training, is legal. And I certainly can't afford a patent lawsuit.

Others have made some of the points I'm trying to make more eloquently. John Carmack, of id Software, said "The idea that I can be presented with a problem, set out to logically solve it with the tools at hand, and wind up with a program that could not be legally used because someone else followed the same logical steps some years ago and filed for a patent on it is horrifying."

I strongly encourage that you read and carefully consider this article:
<http://www.groklaw.net/article.php?story=2010092621054289>

It expresses an important point - that just because a particular code can be executed on a physical device, the code itself is not a part of that device. The code is independent of the device, and exists as an abstract idea. In fact, software developers often strive to write code in such a way that it will easily run on many different devices in order to facilitate reuse of that code. It is also desirable to design code that is independent of a particular application.

For more proof that code is independent of a particular machine, consider an emulator (<http://en.wikipedia.org/wiki/Emulator>). An emulator is software that implements a physical machine so that software designed for that machine can be executed on the machine that the emulator was designed for. In essence, the machine itself can be replaced with software run on an entirely different machine.

For these reasons, the USPTO should reject software patents, as all software is composed of abstract ideas, and, though executable by machines, software is not a part of or tied to any particular machine.

All software can be executed with a pen and pad of paper.

Thank you for your time and consideration.

Respectfully,

Dustin M. DeWeese