

From: David Bruce [e-mail redacted]
Sent: Saturday, September 25, 2010 3:26 PM
To: Bilski_Guidance
Cc: [e-mail redacted]
Subject: Software should not be patentable

Hello,

My name is David Bruce. By profession, I am a 49 year-old transplant surgeon at the Ochsner Clinic in New Orleans. However, I am also an independent software developer, the creator of educational software that is widely used by children, parents, and teachers around the world (<http://tux4kids.alioth.debian.org>). I have been writing "production" software code for over ten years. I believe that software patents are inappropriate, and are becoming a rapidly-worsening obstacle to software developers.

The basis for patents is given in Article I, Section 8 of the United States Constitution:

"To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries;"

Thus, for a patent to be justifiable, it must "promote the progress of science and useful arts". In the context of software development, patents should be allowed if they result in an increase in society's ultimate access to useful software, and disallowed if they serve primarily as obstacles. Importantly, the goal of a patent system is not to ensure profits for patent holders, only to promote progress. I submit that software patents, as currently practiced, have very little to do with the promotion of progress, and everything to do with setting up "toll booths" to collect revenues from the industry.

The current U.S. policy of allowing patents on software is both a geographic and historical aberrancy. The U.S never allowed software or business method patents until the State Street decision, and most of the rest of the world excludes software from patentability. Generally, algorithms and mathematics have never been patentable, and so many people object to software patents on the basis that "all

software is mathematics". To computer scientists and mathematicians, such a statement is beyond dispute. Indeed, it is mathematically impossible for software to be anything but mathematics. Many patent lawyers will deny this vigorously, but it is not a point in dispute by anyone who actually has a working knowledge of computation theory.

This summary from a computer scientist (<http://www.groklaw.net/articlebasic.php?story=20091111151305785>) should be required reading for anyone involved in public policy regarding software patents.

However, my own argument against software patentability is less technical - software code is really just a specialized form of writing, and written expression is not patentable. The lay public does not really understand much about programming. They tend to think of programs as "things", in a way they do not consider novels or musical compositions to be "things". From such a viewpoint, it may seem plausible to speak of a software "invention" that might merit patent protection from independently-written programs that do similar things. However, to someone like me who actually writes software, such an idea is absurd. Software authors are appropriately and justifiably protected from unauthorized copying of their work by copyright law, just as the authors of novels are protected from the creation and sale of unauthorized copies of their works. Patents, however, prevent not only unauthorized copying, but the independent creation of similar works when no copying takes place. The public would never allow Ian Fleming to be granted the exclusive right to publish novels about secret agents, or that only J.K. Rowling can write about young wizards and witches. Similarly, imagine if patents were granted on musical concepts such as the fast-slow-fast sonata form, or the twelve-bar blues progression. As preposterous as such concepts seem, it is no exaggeration to state that today's programmers are in an equivalent situation.

Current software patents do not cover single programs. Rather, they cover common concepts that are widely used in all programs of non-trivial complexity. It is not possible for programmers to be aware of the tens of thousands of patents that could potentially be infringed (at least arguably) by a major software project. Software patents are not supposed to be granted for concepts that are obvious to those

knowledgable in the field. However, this "requirement" is belied by the fact that it is possible for programmers to inadvertently infringe numerous patents without even being aware of the existence of the patent, its holder, or the patented "invention".

Patents are to protect inventors from getting their "ideas stolen". How can a patent holder claim that his/her idea was "stolen" when the "infringer" never even heard of the patented "invention"?

In the last few years, patents are increasingly held and enforced by "Non-Practicing Entities" or NPEs, colloquially known as "patent trolls". These are companies that offer no products, existing solely to acquire patents and threaten companies. These notorious companies are progressively becoming a "cost of doing business". Even if the said patents are dubious, it is much cheaper to pay a "licensing" fee than to pursue a costly legal battle to invalidate the patent. Again, how do such companies "promote the progress of science and useful arts"? These companies have simply found a lucrative flaw in the system, and are exploiting it to great profit.

Software patents do not serve the goals intended for the patent system under our constitution. I implore the USPTO to agree with most of the rest of the world, as well as its own precedent prior to State Street, that computer software is excluded from patentability.

David Bruce, M.D.