

From: Jason Self [e-mail redacted]
Sent: Friday, September 24, 2010 2:31 PM
To: Bilski_Guidance
Subject: Stop Issuing Software Patents

I understand that, normally when the USPTO solicits feedback like this, they hear almost exclusively from patent attorneys who have a vested interest in making sure that patents are granted as broadly as possible. I don't think that you can really rely on what they have to say, even if they claim to be neutral. Their vested interest presents a conflict of interest, in my opinion, and they can't be relied upon.

In comparison, I am just a member of the public.

Patents that read on software should not be allowed. It comes to that: It shouldn't be allowed.

One method to used to circumvent the limits of software patentability, is to claim not "software" but "software and a computer". The goal is to present one non-innovative object (the computer) and one non-patentable object (the software) and get a patent on the combination. The argument made is that, when the software is put on the computer, the computer becomes a "new machine".

One example of this logic being rejected by the US CAFC (appeals court), is the in re Alappat decision, which said: "As the player piano playing new music is not the stuff of patent law, neither is the mathematics that is Alappat's "rasterizer.""

If we have patented an automobile which can drive anywhere, we cannot then come back and file patents for driving from Albuquerque to San Diego, etc. -- the more general patent already applies.

The general purpose computer, which can perform any computation, is not new. Filing additional patents on particular subsets of that general computing ability of the computer is like filing for a patent on driving a car from one particular place to another -- it is a subset of the capabilities which have already been patented. It is in fact simply filing for a patent on a table of numbers which happens to make a given computer perform a given computation -- yet the general property of being able to perform any computation which can be expressed as a table of numbers has already been patented. Thus any patent on software is redundant.

In addition, I raise the following points:

Software Patents Reduce Innovation

=====

Most software innovation happens through leap-frogging: company A comes up with an idea, company B replicates it with extra features or improvements,

company A improves it further. This process is in the interests of innovation and in the interests of the consumer. It expands the market, and very often both A and B benefit from it. Patenting can only slow this process down, to the detriment of the consumer, the market, and the companies who supply that market.

Very often the first company with an idea doesn't get it quite right, or fails to realise its true potential. Their product fails (or doesn't succeed as much as it could have) because they execute it badly or market it badly.

Another company then builds on the idea and succeeds where the first company failed. (Example: the Wang object technology patents, acquired by Kodak after Wang failed, versus Sun. Should the patent system reward failure?)

Not only are there studies showing that patents are blocking innovation and research, but there is also decades of proof that software progress happens without patents.

According to the 2008 Berkeley Patent Survey: Among software companies, the results are even more striking, with them reporting that patents provide less than a "slight" incentive.

http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1429049

* Microsoft Windows 95

Microsoft DOS and Windows 95 are two examples. In 1995, Microsoft had only 77 patents. After Microsoft attained a dominant market position, they started saying patents were necessary for software development, but they actually wrote their software before they started getting patents.

* GNU/Linux

Free software such as the GNU/Linux and FreeBSD operating systems which were developed without patents. 91% of the top 500 super computers run GNU/Linux.

* The WWW and email

The World Wide Web is another example, and email is another.

So software doesn't need patent protection.

In addition to blocking innovation and benefiting from protection,

Software patents block competing software, reducing choice

=====

The evidence suggests software patents are used strategically; that is, to prevent competitors from developing in a similar field, rather than to incentivise innovation.

As just one of many examples, Akamai attacks Limelight as well as other

competitors:

<http://arstechnica.com/tech-policy/news/2008/03/akamai-takes-another-software-patent-scalp.ars>

Software patents enable control of entire markets

=====

Due to the speed at which computer users expect software features to become standard, and due to the need for compatibility, software domains are particularly vulnerable to being held to ransom by the patents of a single company. As just one of many examples: MPEG-LA claims that there is now no way for anyone to write any video player without infringing their patent portfolio.

Finally, and most importantly:

Software Patents Block Freedom

=====

Software development, like writing a book or writing music, is something anyone can do. Most people will never write a book or any substantial amount of music, but some people do, and society is enriched by the work of those people. It's also good to know that if, one day, you really do want to write a book or some music, there's nothing stopping you. No one will send you a letter claiming they have a patent on an idea that you've illegally used in your book.

Similarly, if you ever want to write software, for yourself or for others, at no cost or for a fee, you should be free to do so without worrying about patent infringement.

Proof that these freedoms are useful: The freedom to manufacture cars is not very practical for individuals. In contrast, the freedom to develop and distribute software has led to the creation of very large software projects including the GNU/Linux operating system and other free software.

From: Jason Self [e-mail redacted]
Sent: Friday, September 24, 2010 5:39 PM
To: Bilski_Guidance
Subject: Re: Stop Issuing Software Patents

Jason Self [e-mail redacted]wrote ..

> Software patents block competing software, reducing choice

> =====

> The evidence suggests software patents are used strategically; that
> is, to prevent competitors from developing in a similar field, rather
> than to incentivise innovation.

>

> As just one of many examples, Akamai attacks Limelight as well as

- > other
- > competitors:
- > <http://arstechnica.com/tech-policy/news/2008/03/akamai-takes-another-s>
- > oftware-
- > patent-scalp.ars

Here's another example of a company using patents to eliminate competitors:

<http://www.engadget.com/2010/09/22/apple-sues-hypermac-battery-maker-for-using-patented-magsafe-and/>

Even if Apple doesn't win, a lawsuit can bankrupt a small company and just the THREAT of a lawsuit can be enough to prevent competitors from trying to enter a market.