From: Richard C Ballantyne [e-mail redacted]
Sent: Friday, September 24, 2010 1:31 AM
To: Bilski_Guidance
Cc: [e-mail redacted]**Subject:** Bilski Ruling and software patent eligibility

Dear USPTO,

As a developer of custom software for Goodrich in Ithaca, New York, I request that you please stop issuing patents for software ideas. It is practically impossible for me to know if the ideas I implement while developing software are already patented. First, my employer does not authorize me to verify that the code I create does not infringe the hundreds of thousands of software idea patents already issued by the USPTO. Second, to my knowledge, my employer does not employ someone to verify that the code I create does not infringe software idea patents. Third, even if my employer did allow me to check if I was infringing software idea patents, it would be impossible for me to do; the rate at which software patents are issued far exceeds the rate at which I can read them. Finally, I am not a lawyer and do not understand the legalese used in many software idea patents. The bottom line is that there is no practical disclosure mechanism for software patents due to their sheer volume.

I have friends who develop software for small software companies. They often complain about the patent minefield they must walk through. Some have received threats from larger companies with huge software patent arsenals. Others are forced to enter expensive cross licensing agreements in order to stay in business. It is prohibitively expensive for them to hire a patent lawyer. A few have just thrown up their arms in frustration and decided to work for a larger company in order to avoid liability. Larger software development companies such as Microsoft, Apple, and Adobe have huge patent arsenal which they can use for defensive purposes. My brother, who is a senior software design engineer at Microsoft and my sister who is a patent lawyer have admitted to me in private that there are serious problems with the patent system especially in regards to software idea patents.

From my experience software patents do more harm to the vast majority of businesses and society than good. This includes businesses that are not in the business of making software. The reality is that:

1) Software idea patents do not "promote the progress of science and useful arts" because they stifle innovation. Software idea patents make it unfordable for SMEs to develop new software safely. A recent example of how software patents stifle innovation are the problems that have arisen for the HTML open standard in adding support for streaming video.

2) Software idea patents reduce the quality of software by allowing companies to push closed file formats which cannot be read or written to using third party software such as Free Software. The bottom line is that computers are harder to

use than they need to be because the software on them is artificially limited.

3) Software is not patentable subject matter since all software that is actually run is always reduced to a series of mathematical operations. It is a fact that mathematical operations are simply "laws of nature" and are thus not patentable. It is a fact that laws of nature can only be discovered and not invented. It is a fact that in software, the exact sequence of mathematical operations is discovered by the compiler the vast majority of the time. Expecting the creators of compilers (or other software that dynamically creates other software) to anticipate future software idea patents and then change their compilers to prevent outputting patented mathematical operation sequences is practically impossible.

4) Software is very different from most patentable technologies. First, software is simply information that takes an intangible form. There is not one patent for one software product, as it is with many physical products. Second, the cost of the tools used to create software is essentially free. Last week I just picked up four computers for free at my local recycling center. I regularly use the Free Software compiler GCC, and Free Software IDE Eclipse to author new software. Just like anyone can author a book using simply a pen and a stack of paper, or record some music using their computer, anyone who can read and do math can also learn how to program and author software using a low cost or free computer. Books, music, and videos are not patentable subject matter and neither should software. Software is simply mathematical information and is already protected by copyright like other forms of information (videos, music, books, etc). Software does not need to be "protected" by patents too.

5) Even the simplest software, such as the "Hello World" program, is built using thousands of different ideas. This is because to create anything useful in a reasonable amount of time, developers use existing software functions stored in libraries (such as a DLL). They often bundle these libraries into their released software. It is practically impossible for developers to know which functions in these libraries already contain patented ideas.

6) The 20 year duration of patents is an eternity in the realm of software. The USPTO should immediately reduce the term from 20 years to 2 to 5 years for software idea patents while it investigates whether or not to abolish software patents.

7) Most software (except Open Source or Free Software) is released in a compiled format -- in machine language that is effectively impossible for humans to understand. Thus many patents cover "inventions" that have been in the public domain for years. However, since the source code which humans can understand is unavailable, it cannot be proven that the patents issued cover "inventions" which are already in the public domain.

8) Many obvious software "inventions" end up getting patented, such as the "one click" patent that was awarded to Amazon, and the "auto-filling" feature in spreadsheets. Is this because patent examiners are experiencing the same frustrations that developers face? There are too many existing software idea patents to wade through in order to determine if a new software idea patent should be issued.

The USPTO can, and should, exclude software from patent eligibility on at least these legal grounds: software consists only of mathematics, which is not patentable, and the combination of such software with a general-purpose computer is obvious. Software patents just drive up costs and frustrate users who are forced to use software that is artificially limited. I foresee rapid advances and improvements in software as soon as the patent office abolishes software patents.

Thank you for giving the public an opportunity to suggest guidance for the USPTO in the wake of the Bilski ruling.

Sincerely,
Richard Ballantyne

(Software Engineer at Goodrich Corporation