-----Original Message-----
From: Bob Frazier
Sent: Friday, January 04, 2013 7:00 PM
To: SoftwareRoundtable2013
Subject: software patent comments

Software patents are inevitable to protect CERTAIN KINDS of inventions, particularly those in which SOFTWARE is a key to the functionality of a device.

However, unless the software is SPECIFIC to that device, patents should not be granted.

Additionally, software itself that is intended to run on MULTIPLE devices by either being installed or bundled with the device (but can be held as "separate" from the device, even if the software's functionality is key to the device) should NOT be covered by a patent.

Example #1: An algorithm that dynamically steers antennas, that is required for a steerable antenna to work, should be patentable FOR THE STEERABLE ANTENNA. The software would be an integral part of the invention, and designed to work WITH that specific invention. And the algorithm itself may not be adequately protected by copyright, allowing a competitor to unfairly use it without patent protection.

Example #2: An algorithm that stores data in a particular format, which is part of a data storage device. This algorithm COULD be used on any number of devices OTHER than this particular data storage devices, and as such, should NOT be patentable. Additionally, a DIFFERENT format could ALSO be used with the same data storage device, indicating that the software algorithm itself is NOT needed for the invention to be viable. As such, if the algorithm is patentable at all, it should ONLY be for THIS particular type of device. Use of this algorithm on OTHER devices that are NOT applicable to other patent coverage for this device should not be in any way prevented.

Example #3: An algorithm for human interaction with a device that allows multiple types of button presses of varying lengths on a single button to make different types of menu selections. This algorithm actually exists for multiple patented devices, from iPods to certain printers, though they appear to ONLY apply to specific CLASSES of devices [iPod vs printer, for example]. Such a "user interface" patent should NEVER be granted, since it CAN EASILY APPLY to a wide variety of devices NOT covered by the patent, EVEN IF IT IS SEEN AS A COMPETITIVE ADVANTAGE to the inventor for a particular class of devices. As it now stands it appears that the printer patents and the iPod patent apply to their specific device classes, but it is still a USER INTERFACE. USER INTERFACES should NOT be covered by any kind of patent. Past decisions regarding Microsoft vs Apple and the Windows interface is a good example of why.

It is also my opinion that patents for data compression methods (like MPEG), file formats (like FAT32), and data interchange (like SMB packets), should also NOT be patentable. Since there is value in licensing a software library to a manufacturer or software provider, the revenue stream still exists. If the software is copyrighted, then it is protected against plagiarism. If the software is worth a patent, then there must be sufficient intellectual property involved SUCH THAT a competitor would have to re-write a software library FROM SCRATCH to prevent violating the copyright, which is a SIGNIFICANT amount of effort. As such, EXISTING COPYRIGHT PROTECTION is sufficient for library providers. If such providers choose to make their libraries "open source", they can STILL restrict the distribution or use of their libraries through the license agreement. For this reason NO patent protection should be applied to any kind of GENERIC software algorithm (including data compression methods, file formats, and data interchange) since an EXISTING library can still be licensed at a significant cost benefit to potential competitors AND for financial gain by the developer.

So, if a software patent is granted, it should

a) be SPECIFIC to the invention, and integrated with it

b) be applicable ONLY to that SPECIFIC invention, so that similar software on DIFFERENT inventions not covered by the patent will not be affected

c) not be "pure software", that is, installable on multiple platforms (such as a library or a program)

d) be something that can NOT be protected by simple copyright

Additionally, the USPTO should look CAREFULLY at the KINDS of litigation that has been taken in the past over software patents. We do not need "patent trolls" interfering in the natural course of innovation in the area of software. If a programmer must perform tedious patent searches every time he writes code, NOTHING WILL GET DONE! It is EXPENSIVE to involve patent searches for otherwise trivial things. This is EVEN WORSE for independent developers and startup companies, who do NOT have a "staff legal team" to go over these things. We all know that PATENTS WERE INTENDED to "protect the little guy". So we MUST keep this in mind with WHATEVER decisions are made on software patents. INDEPENDENT DEVELOPERS and STARTUP COMPANIES must be protected FIRST. That is the INTENT of the patent system.

On a related note, I do both software AND hardware design. I have had to spend a number of hours on the uspto.gov web site doing patent research to prevent "using someone else's idea". The devices I have been working on are PROBABLY better covered by copyright, since they are
(literally) hardware+software solutions, and so the software would be copyrighted, and licensed for use with specific hardware. The software is what makes the hardware work (without the software, the device is a "brick"), so in theory it COULD be covered by a patent, but due to its overall complexity, a copyright makes more sense. Even though the hardware is 'not unique', the hardware + software is. But if a competitor DID write his own software, it would probably take him as long as it took me, and it would probably be worth licensing MINE instead.