

From: Aaron Denney [e-mail redacted]
Sent: Monday, September 27, 2010 2:21 PM
To: Bilski_Guidance; [e-mail redacted]
Subject: scope of software patents

Undoubtedly there will be many comments emphasizing the bad effects of software patents, and that for policy reasons they should not be accepted. While I agree with this argument, I see other fundamental problems with applying patents to software, and write to discourage their use.

I instead want to stress that software is not a machine, but instructions, an algorithmic, step-by-step, description of processes. This core nature of software means that even when the process being described by the software is patented, this should not hinder the distribution of the software. A patent application is itself much the same thing -- a description of the process to be covered. It would be utterly ludicrous to forbid the distribution of an approved patent application for violating the patent. The description is not the process itself, nor is it a machine for performing the process. The same really holds true programs. Imagine taking a patent application and annotating with details of a particular way of implementing the patent. At no point does it make sense to forbid the distribution.

It is, of course, perfectly reasonable to have an otherwise patentable invention be partially implemented in software. I must argue though that what violates the patent is the whole machine (including the software, to be sure), and not the software itself, nor the general purpose computer itself. Either should be free to be built, used, sold, and otherwise distributed. This is not much different from parts in an invention being illegal to combine in a way that violates the patent, while perfectly legal to have apart or combine in other ways.

I will also note that in practice, many software patents have been granted that seems obvious. Combining something with a general purpose computer should never have been considered non-obvious, nor should doing something with a computer network as an intermediary be considered non obvious. Computers were built to perform algorithmic processes.

Selecting a particular algorithm may not be obvious, but the use of a computer to execute it certainly is obvious. Computer networks were built to carry general information; making that type of information more specific to a certain use does not make the use of a general network less obvious. Many machines are adapted to a specific use, and must be readapted to be used in a different, but similar circumstance.

Computers and computer networks are different. They are general purpose machines where the general really does encompass all specifics.

--

Aaron Denney

-><-