

From: adrian stovall [e-mail redacted]
Sent: Sunday, August 01, 2010 2:52 PM
To: Bilski_Guidance
Subject: Comments...

I work day-to-day as a software developer for a commercial consulting firm. As such, I have a vested interest in the way in which the USPTO considers patents related to software.

The machine-or-transformation test has been problematic for software because of the way the USPTO has used it in the past.

The Bilski decision is the first acknowledgment by a US court that software is not patentable in and of itself.

Computer software for modern computers is software for generic computing devices. This statement is true for computers ranging from federally-funded supercomputing clusters to commodity servers to desktop and laptop PCs to mobile phones. All of these are general-purpose computing devices, and all can perform substantially the same operations (at different speeds, screen resolutions, and fidelity). The vendor of the machine is unimportant, as is the specific operating system, compiler, or programming language used.

Software is necessarily abstract. In all instances from music boxes to weather simulators, software is a set of instructions for the computer...an algorithm. A specific input **always** produces a specific output, and the relationship between input and output is discovered, not invented (as is true for all algorithms). Even in the case of a music box, it is not the unique pattern of raised bumps on the cylinder that is novel, but the physical implementation of a machine that turns them into sound. Software never produces a physical effect...it directs physical hardware to produce the effect for which it was designed.

Thanks to Bilski, we have courts taking note of this distinction, and the USPTO should, as well. Allowing the issuance of patents for software stifles innovation and expressly prohibits the advancement of the useful arts as related to software. No software developer can be certain that any piece of software is conceptually unique in every

detail, and in most cases, developers may be certain that they are going to be running afoul of existing patents simply by writing a program that produces meaningful output.

Until software is ruled to be no longer patentable by the USPTO, and existing software patents are ruled invalid, I will be unable to be certain that the unique works that I and my fellow developers create will not lead myself, my employer, or our clients towards an expensive and unnecessary lawsuit.

In short, patents and software do not belong together, and cause substantial harm to all software development efforts.