From: edwin hunt [e-mail redacted]
Sent: Monday, September 27, 2010 6:02 PM
To: Bilski_Guidance; [e-mail redacted]
Subject: Input re: Bilksi decision

As a college student and future software developer, the recent Bilski v. Kappos decision was of particular interest to me. The Supreme Court's ruling that the machine-or-transformation test should not be the sole test in defining patentability may mean that when I graduate and begin writing software for a new generation of computers, handheld devices and websites, I may not have to face the same legal landmines that have trapped many developers prior to this decision.

Software patents today are incredibly broad in their application because unlike patents submitted in previous generations, a software patent does not actually have to describe how a process or transformation will happen. Instead, it merely declaims that it must happen. Had Henry Bessemer applied for a patent that covered "a low-cost method for creating steel from pig iron, using hot air, coke and trace elements," he would have never have been granted patent #16082. In the unlikely situation that such a broad patent had been granted, it would have also have covered the Siemens-Martin process, because the patent does not actually show how the transformation from iron to steel occurs. If the entire steel factory were contained within a giant black box, to an outside observer the transformation in both cases would be largely the same. This is despite the fact that the processes themselves are internally vastly different.

Software is just such a "black box." To an outside observer, the functionality of two pieces of software may be largely the same. As long as the solution is non-obvious the methods of accomplishing that functionality can be myriad, yet software patents, unlike any other type of patent, allow and in fact encourage this sort of black box description!

Bilski v. Kappos explicitly denies a process that, had it been applied for in the context of software and a computer, and without details of the mathematical formula for risk minimization, mimics patents such as Amazon's 1-click and Apple's travel services patent application #20100190510. If such a specific process as Bilski's could be denied, then more generalized processes such as Amazon's and Apple's, differing chiefly in their combination of combining software with a previously existing computer, cannot stand under the same test.

The Supreme Court's ruling on Bliski v. Kappos, if interpreted to narrow patent law, would mean that future generations of software developers, so long as the methods used within the software are different, would no longer be subject to the corrosive effects of "patent trolls," litigation threats from major corporations, or overly vague patents stifling innovation and commerce. It is my sincere hope that by the time I am ready to begin contributing to the American economy, these problems will no longer present a hazard to software developers attempting to offer new or better products to both fellow Americans and foreign consumers. As the alternative is an inevitable slowing of innovation within the United States as our own patent system makes overseas software development less dangerous and easier, removing or narrowing the protection of patents on software is not only beneficial to developers, but also to the nation as a whole.

v/r,

Edwin Hunt