Regarding *In re Bilski*

Ben Klemens

31 October 2008

The ruling for *In re Bilski* is out. We had all hoped that it would resolve some longstanding questions on the patentability of software, but instead it is merely the prelude to a debate.

This was an *en banc* ruling, where the patent on a business method was struck down 9-3. That alone shows us that the Court of Appeals for the Federal Circuit is not a bunch of pro-patent fanatics—especially the majority that aren't former patent attorneys. The judges who do still believe that everything should be patentable are in a dwindling minority. They don't have support at the Supreme Court, they don't have support at the Patent Office, and they don't have majority support in their own building.

OK, enough politics; on to the content of the ruling. The primary contribution of the ruling is to sweep away much of the cruft that has grown around this topic over the last decade, including several ad hoc inventions of the very court that wrote this ruling. It arrives at a simple rule for determining whether something is patentable: is the patent claim for a machine or transformation of matter? If so, then it is patentable subject matter. The overall ruling seems to congratulate itself over sweeping away the cruft and getting to that simple, clear heart of the matter.

But it doesn't answer the question. All the cruft (which I'll discuss further below) evolved as many separate parties tried to answer the question of what is a machine. It's not an easy question. For example, a mental process, which is definitely not a machine, could easily be written down to a hard drive, which definitely is a machine. Is the data-plus-drive amalgam a machine or just a mental process with an off-the-shelf part added to get around a technicality? That's really the billion-dollar question: when does an abstract algorithm become a machine?

Here's the answer from the ruling: "We leave to future cases the elaboration of the precise contours of machine implementation, as well as the answers to particular questions, such as whether or when recitation of a computer suffices to tie a process claim to a particular machine."

So the ruling explicitly states that it won't answer the key, central question.

On the positive side, the ruling does make progress. First, in throwing out all that cruft, it threw out many of the excuses that people give for software being patentable. Notably, it rejects the assertion that any process of any sort can be patented. The machine-or-transformation rule requires that the process either be fixed in the form of a certain machine, or make a transformation beyond just multiplying numbers together.

Despite claiming that all that matters is the machine-or-transformation test, the ruling also bears in mind many other necessary conditions for patentability, such as the rule that a patent may not "wholly pre-empt" a law of nature or principle or mathematical formula. Also, if you wholly pre-empt a mathematical algorithm within some narrow field of endeavor, the court rules that this is still a pre-emption. I'll have a little more on this below.

There is the *State Street* rule that an item is patentable if it produces a "useful, concrete, and tangible result," which this ruling explicitly dismissed. I always hated this rule, because it is entirely vacuous, especially as the *State Street* court took pains to interpret *concrete* and *tangible* broadly—and it's hard to imagine a patent that is worth filing but is somehow not useful. If your patent draftsman can't write up your patent to be useful, concrete and tangible, it won't be hard to get a new one who can. But we now need to determine patent-eligible subject matter by the machine-or-transformation test presentend in this ruling, which is not vacuous, so *State Street*'s blank check for any process patent has been revoked.

The ruling pointed out that U.S. law has never had a "technological arts" test for patentability. They did the right thing here, because the phrase is impossible to define precisely. We're having enough trouble defining the word *machine*; there's no benefit to shifting the debate to the more vague and subjective word *technology*. Similarly, there's no point in defining an exception to patentability like *business methods*, the ruling states, because it'd be a waste of time to try to pin down a definition of the excluded field.

On the key question of when information becomes a machine, the ruling does provide some clues: "First, [. . .] the use of a specific machine or transformation of an article must impose meaningful limits on the claim's scope to impart patent-eligibility. [. . .] Second, the involvement of the machine or transformation in the claimed process must not merely be insignificant extra-solution activity." [For more on extra-solution activity, have a look at my law review article[1] that focuses heavily on the idea.]

In the earlier segment on pre-emption, we got what I take as a sample application of the first of these rules: in the algorithm in the Supreme Court's *Benson* case ". . . the limitations tying the process to a computer were not actually limiting because the fundamental principle at issue, a particular algorithm, had no utility other than operating on a digital computer. [. . .] Thus, the claim's tie to a digital computer did not reduce the pre-emptive footprint of the claim since all uses of the algorithm were still covered by the claim." The ruling gives other examples, many taken from the Supreme Court, of cases where the computer does or does not limit the algorithm in a sufficient manner to restrict patentability (notably in *Flook*, where an alarm to measure hydrocarbon temperature is still not sufficiently tied to the machine to merit patentability).

The Benson example, which talks about limiting scope and pre-emption, and the machine-or-transformation rule regarding how "the use of a specific machine . . . must impose meaningful limits" bear much in common. They indicate that if a computer is merely being used as a transparent medium for the algorithm, then the patent is an attempt to patent and wholly pre-empt an abstract algorithm, and is not a claim on a machine. If the computer is not merely a transparent medium and is a limited use of the algorithm, then the patent is for a machine.

[1] `http://www.bu.edu/law/central/jd/organizations/journals/scitech/volume141/documents/Klemens.pdf`

Did my rubric in the last paragraph sound vague? As stated, it is, and it's all this ruling leaves us. We have moved a little forward in the debate, because the ruling has thrown out things like the "useful, concrete and tangible" rule and done what it can to shut down the argument about a technical arts test. It thus allows us to focus a little more closely on the question of when an abstract work becomes sufficiently fixed to a physical environment that we can call it a new machine. It provides a few hints about how that measurement should be made, in the form of a repetition and not-overly-narrow interpretation of pre-emption rule and an emphatic repetition of the rule from two Supreme Court rulings that extra-solution activity does not make something patentable. But the key question itself remains unanswered and still open for debate.