Andrew Tridgell on Patent Defence How much can programmers gain from patent literacy?

Hartmut PILCH

http://eupat.ffii.org/10/03/tridgell

May 14, 2010

The Samba project has been afflicted by patent claims. Microsoft has been threatening this project, whose aim is to allow different operating systems to interoperate. In doing so, the Samba project is forced to use a proprietary standard of Microsoft that is encumbered by patents.

Andrew Tridgell has recently given a lecture in which he teaches programmers some basics of the patent system. He nicely illustrates some of the misunderstandings that occur when programmers, lawyers and patent attorneys try to communicate. However there is a risk that, if people apply his teachings too narrowly, they might fall prey to misunderstandings of the same kind.

Andrew's main message is:

0. Read patents
1. You don't have to care about dependent claims
2. If you're not using ALL elements of an independent claim you can't be infringing the claim in question.
3. If we managed to do make patent claims more transparent, this could allow programmers to stay out of harmsway.

The last item seems far too optimistic, because the biggest problem is not a lack of transparency of patents but their existence.

Andrew's audience might be inclined to underestimate the exclusion power of patent claims.

Patent claims are meant to be real. You can usually not get around patent claims by playing word games, e.g. by presenting your solution in different terms.

The most important theoretcical tool by which patent courts ensure that patent claims have real exclusionary power is the Doctrine of Equivalents.[1]

As the U.S. Supreme Court said in GRAVER MFG. CO. v. LINDE CO., 339 U.S. 605 (1950):

> "One who seeks to pirate an invention, like one who seeks to pirate a copyrighted book or play, may be expected to introduce minor variations to conceal and shelter the piracy. Outright and forthright duplication is a dull and very rare type of infringement. To prohibit no other would place the inventor at the mercy of verbalism and would be subordinating substance to form. It would deprive him of the benefit of his invention and would foster concealment rather than disclosure of inventions, which is one of the primary purposes of the patent system."

What this all boils down to is that you indeed have to read (and understand) all claims, consider what the patented invention does as a whole and evaluate if your solution achieves the same.

If it does, you are very likely infringing even if your solution does not use all claim elements. That's one of the reasons why coexisting with software patents is a more complex task than Andrew's talk suggests.

[1] The doctrine of equivalents (Äquivalenzlehre) has traditionally been very strongly developped in Germany. Even if at the moment it is being applied with more restraint, it is there to stay and to come back under various namings (e.g. it used to take the form of a theory of the "general inventive thought" (Allgemeiner Erfindungsgedanke) before the EPC came into force). Moreover, the main motive of the recent restraint was the need to temporarily curb a swelling flood of patent applications by simple formalistic means. But in the wake of the economic crisis the number of patent applications has dropped.

[Note: this text is based on contributions of members of the eupat mailing list.]