THE TRESPASS FALLACY IN THE “SOFTWARE PATENT” DEBATE

Ryan T. Holte*

In *The Trespass Fallacy in Patent Law*, Professor Adam Mossoff details how patent law jurisprudence and scholarship is dominated by an indeterminacy critique or “trespass fallacy” in two respects. First, describing conceptual error, Professor Mossoff details how judges and academics have improperly conflated the entire legal rights of patent title to the single legal doctrine of property law trespass. Second, focusing on empirical error, Professor Mossoff describes how the indeterminacy critiques of patents utilize only an idealized theory of how trespass is thought to function, without formal empirical data regarding how trespass or other real property boundaries actually function within litigation. Professor Mossoff’s essay makes an important contribution to patent law scholarship by breaking through this improper rhetoric and exposing the misleading and unverified indeterminacy critiques of patents for what they are—trespass fallacies.

Professor Mossoff’s essay, however, only briefly mentions the now paramount contemporary issue surrounding the more-focused “software patent” debate. In this short essay, I will briefly discuss Professor Mossoff’s trespass fallacy analysis as it relates to “software patents” and the Supreme Court’s October 2013 Term case *Alice Corp. Pty. Ltd. v. CLS Bank Int’l*. Since the phrase “software patent” lacks a clear objective definition, I will use the more technically accurate—albeit less rhetorically thrilling—phrase “computer-implemented inventions.”

* Ryan T. Holte, Assistant Professor of Law, Southern Illinois University School of Law. © 2014 Ryan T. Holte. Comments welcome at rholte@law.siu.edu.

2. Id. at 1690–92.
3. Id. at 1692.
4. See id. at 1690.
7. The term “computer-implemented invention,” is the language used by the Supreme Court in the Question Presented for the *Alice* case: “Whether claims to computer-implemented inventions—including claims to systems and machines, processes, and items of manufacture—are directed to patent-eligible subject matter within the meaning of 35 U.S.C. §101 as
While I cannot address every issue in the policy and legal debates over computer-implemented inventions in this short essay, what is clear is that the debate is, thus far, rife with trespass fallacy.

Much has been written regarding the increased number of computer-implemented invention patents and the increased litigation surrounding these patents. This scholarship often includes an indeterminacy or “vagueness” argument against computer-implemented inventions. The arguments are especially pronounced regarding the § 101 patentable subject matter issues involved in the now-pending Alice case. Numerous amicus briefs filed in Alice argue against patentability of the Alice Corporation patent due to problems of “vagueness” generally. For example, Google argues “[s]oftware patents . . . ‘raise special problems in terms of vagueness’ . . . without a ‘high enough bar,’ ‘patent examiners and courts could be flooded with claims that would put a chill on creative endeavor and dynamic change.’” It was perhaps unsurprising that questions asked by the Justices at oral argument touched on legal requirements to “sufficiently” describe an invention and whether this was possible for computer-implemented inventions.

This assertion of “vagueness” in patents covering computer-implemented inventions is widespread outside the Alice case. For some time, academics have argued that “[s]oftware patents are overbroad” or “fuzzy” compared to other patents. Some now contend that

---

11. Transcript of Oral Argument at 39, Alice Corp. Pty., Ltd. v. CLS Bank Int’l., No. 13-298 (U.S. argued on Mar. 31, 2014) (question by Justice Kagan), available at http://www.supremecourt.gov/oral_arguments/argument_transcripts/13-298_869d.pdf (“Mr. Perry, before we get back to these matters, you said to Justice Scalia if a patent sufficiently describes how a computer will implement an idea then it’s patentable. So how sufficiently does one have to describe it?”).
defendants accused of infringing computer-implemented patents need “a mechanism that (1) can keep as many broad software patents as possible from leaving the PTO in the first place and (2) they can use to attack broad software patents that slip through the cracks before trial with an early motion.”

The arguments that computer-implemented technology patents are “vague,” “overbroad,” and “fuzzy” seem to reflect the trespass fallacy identified by Professor Mossoff, at least in their framing or foundational assumptions. More specifically, these critics seem to be invoking the “nirvana fallacy” of optimal patent clarity that can never be obtained—they are assuming some idealized and empirically unproven baseline of clarity in patents by which they are definitively proving that computer-implemented inventions fail. But this comparison is not possible. There is no empirically proven standard or even established optimal rate of valid patents by which “clear” or “non-vague” computer-implemented technology patents can be evaluated. Beyond this implicit assumption of the nirvana fallacy, some scholars invoke the trespass fallacy more explicitly in their critiques of computer-implemented inventions: “But where land boundaries are clear and searchable, the boundaries surrounding software patents are notoriously fuzzy.” As Professor Mossoff explains, it is both conceptually and empirically invalid to compare entire patents to physical boundaries for land.

While Professor Mossoff does not explain why trespass fallacy arguments exist generally, the more narrow debate over computer-

at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2139146 (arguing fuzzy software patent boundaries have “been responsible for a significant portion of the increase in patent litigation we have observed since the early 1990s.” and, “[P]olicy makers should adopt reforms that make software patent boundaries more certain for owners and technology users prior to litigation.”).

14. Lemley, supra note 8, at 940 (“Bessen and Meurer are surely correct that patents suffer from notice and boundary problems and that software patents suffer more than most.”).


16. Mossoff, supra note 1, at 1692 (“The empirical error is that there are no formal empirical studies of how trespass or other real estate boundaries function in litigation; thus, the indeterminacy critique uses only an idealized theory of how trespass is supposed to function as an alleged empirical standard of comparison in evaluating the efficiency of the patent system. Economists have long identified this improper comparison between idealized theory and empirical reality as a ‘nirvana fallacy.’

implemented inventions provides some possible insight into this question in two ways. First, the debate about “software patents” lacks any clear standard perhaps because the term “software patent” itself lacks any settled definition. Indeed, there is no legal definition for the term “software patent” used by courts and scholars. This is similar to the policy discussions in my own work regarding “patent trolls.” Even more important, the Patent and Trademark Office does not have a specific classification for “software” patents. If the basis in terminology of legal and policy debates regarding “software patents” are undefined and amorphous, then it is not surprising that the accusations of “vagueness” reflect the same ill-defined and unproven assumptions that comprise the trespass fallacy.

Second, computer-implemented technology is recent compared to other types of technology, such as mechanical or pharmaceutical innovations. Widespread patenting of computer-implemented technologies did not occur until the mid-1990s; courts and the Patent Office are still adapting through experience and tests of patent prosecution terminology to better accommodate the challenges in this new technological area. Computer-implemented inventions even lack proper understanding generally due to their nascent position within the
scientific community. For this reason, “vagueness” issues are likely more normal when dealing with new advances in technology. By way of example, these issues are similar to inventions related to pharmaceuticals and chemicals prior to the introduction of the standardized periodic table in 1869. Once described as broad and not-patentable, chemical patents are now used in comparison as a benchmark of clarity against computer-implemented invention patents, but it has taken over a century of standardization, research, and patent drafting to get there.23

As Professor Mossoff details in his essay, the indeterminacy critique of patents that rests on the trespass fallacy should be jettisoned and replaced with empirically grounded and proper conceptual property law analyses. Nowhere is this more immediately relevant than in the debate over patents on computer-implemented inventions. Hopefully, the Supreme Court in Alice will see through the trespass fallacy and regardless of how it decides the case, will reach a conclusion that is based in properly proven and legitimate empirical and doctrinal grounds. If not, the debate over computer-implemented inventions will certainly continue, whether in the courts, Congress, or both.

---

23. See Petra Moser, Why Don’t Inventors Patent? 4 (Nat’l Bureau Of Econ. Research 13294, 2007), http://www.nber.org/papers/w13294.pdf?new_window=1 (“In 1869, the publication of the periodic table introduced a research tool that greatly facilitated chemical analysis. Exhibition data show that inventors’ propensity to patent chemicals increased substantially in response to this change. In 1851, none of the U.S. innovations in chemicals had been patented. By 1893, the share of patented innovations increased to 16 percent and to 18 percent in 1915. By the end of the 20th century, chemicals had developed into the most patent-friendly industry.”). With clear definitions of terminology (and elements), chemical patents are now described as “precise” and “clear.” Miller, supra note 13 (manuscript at 4) (“In contrast, the scope of patents with more precise structural language, such as those claiming specific chemical compounds, will be better known by disputing parties prior to litigation. Put another way, a chemical compound is a uniquely identified structure and over time the terms used to describe it change slowly or not at all. Conversely, the meaning of many common software patent terms change quickly as new applications are rapidly developed.”); Lemley, supra note 8, at 930 (“A related problem is the uncertainty associated with the meaning and scope of a software patent. Unlike chemistry and biotechnology, where we have a clear scientific language for delineating what a patent claim does and doesn’t cover, there is no standard language for software patents.”).