

Is the Abstract Idea Exclusion A Big Mistake for Software, or is There Something Important To It?

Article By:

Steven Lundberg

As recently reported by [Dennis Crouch](#), under Dir. Iancu, eligibility rejections have dropped to about 6.6%, from approximately 8% before that. Software patent practitioners are pleased to see the needle moving in the “right” direction, but still there are far too many worthy inventions being denied by the USPTO, the PTAB, and ultimately the Federal Circuit. In particular, inventions based solely on highly innovative and 100% technical, electrical engineering subject matter, continue to be treated as if they were claiming methods for practicing voodoo. Let me add that many if not even most of these rejections are perfectly justifiable based on the most recent Supreme Court 101 jurisprudence, and the Federal Circuit jurisprudence attempting to implement those flawed decisions. And therein lies the crux of problem currently faced by the patent system in dealing with the eligibility of many software-related inventions: both the public policy and the logic behind this jurisprudence is a disaster.

I was hoping that eventually someone would come forward with a clear way to analyze inventions to determine if they are “abstract” or “not abstract.” But so far, no one has done it. I have tried to puzzle it through to a logical conclusion, but have not gotten there yet, or perhaps not even close. But I do have some observations that help me understand the nature of what is “abstract” or “not abstract.”

First, I believe that a large majority of inventions submitted for patenting at the USPTO start out their existence, and end their existence, purely in the human imagination. I say a large majority and not all, because in some cases no doubt an invention is physically created simultaneously with its emergence in the inventor’s mind, such as a mechanical invention made possible due to the placement of an additional part in a combination, where the part is physically positioned simultaneously with the inventor’s decision to try it there. But certainly very often, inventions are only imagined by an inventor and the only existence they obtain is perhaps being described in a patent application or inventor’s notebook.

Are these inventions that exist only in the imagination (even when documented they still only have meaning when imagined by the reader) “abstract” so as to not be patent eligible? Of course we know an invention, even if never “actually reduced to practice” is considered “constructively reduced to practice” if it is described in a patent application. So, this cannot be what is meant by “abstract”, in and of itself. So, it is possible for an invention to only exist in an “imagined” form, not be “abstract”, and still be patent eligible. But it is also possible that an invention that is only “imagined” can also be

“abstract” under Section 101.

In a different realm of consideration, I often like to entertain the idea that a novel musical score is a good example of what we could consider “abstract” under Section 101. More particularly, we know most any good patent attorney could draft a patent application that described a “novel” computer configured to generate sound energy in accordance with the musical score, using a computer program. By traditional eligibility logic applied to computer-implemented inventions, this would constitute a uniquely configured computer, and certainly the computer as a whole in combination with the unique program, are as “technical” as you can get. But, most practitioners would never have even attempted to patent a new musical score in the form of a programmed computer, absent some other engineering innovation that was not limited to the novel score. Why? Because, this hypothetical “invention” is not technology, but rather the work of a composer. Note, strictly speaking, this “invention” is no more abstract than any other software invention that effectuates novel operation of a computer — it is just a different type of functionality — one that is not scientific in nature by historical conventions.

On a related note, consider the printed matter doctrine — e.g., see *In re Gulack* (Fed Cir 1983), which holds that printed matter is not patentable unless it has a “functional relationship to the substrate.” In essence, in the example above, the novel subject matter, the musical score, does not in and of itself have an “functional relationship” to its “substrate – i.e., the computer. Accordingly, the computer cannot be said to be “part of” the invention.

So, for today, I will leave you with this idea: could it be that “abstract” under 101, as applied to software, mean at least in part, that the novel aspect of the invention at hand, has no functional relationship to the computer hardware itself, such that the hardware is not fairly part of what is being claimed? This is not to say that there is not technical subject matter being included in such a claim, but if that technical subject matter is not truly functionally integrated with the novel subject matter, perhaps the novel subject matter is indeed merely “abstract” under the printed matter doctrine. Could it be that some computer-related inventions are nothing more than novel arrangements of information more akin to a work of authorship than a technical invention, and that the computer only, essentially, “plays” that information the way a computer might play a new song using a music app? Or is the computer just functioning essentially as a calculator, performing mathematical operations that are not functionally related to the calculator itself.

Please understand these thoughts are by no means claimed by me to be worthy of application in practice, so I am not arguing for their adoption. I am merely trying to better understand if there is any solid, logical basis, for excluding some types of software-implemented “inventions” from eligibility.

© 2025 Schwegman, Lundberg & Woessner, P.A. All Rights Reserved.

National Law Review, Volume VIII, Number 217

Source URL: <https://natlawreview.com/article/abstract-idea-exclusion-big-mistake-software-or-there-something-important-to-it>