Alice Fallout

Last June the Supreme Court issued a decision on software patents called the Alice decision (for a summary of the decision, please see my prior post). The main thrust of this decision was that simply computerizing an otherwise well known activity could not, by itself, be patented. The basis for this decision was that a software algorithm to automate these sorts of processes was too abstract to receive a patent. My initial reading of this result was that the Alice ruling was setting the stage for confusion. In theory at least, the ruling held that software could be patented, in practice the Supreme Court was leaving it to the lower courts to determine the circumstances under which software patents would be appropriate.

The early results of this ruling have been harsh for the holders of software patent holders. In the months since the Supreme Court's Alice decision, there have been at least 10 software patents that have been invalidated by Federal District and Appeals Courts in decisions directly related to the Alice decision.

As expected, there are variations in these rulings but some early patterns do appear to be emerging.

Pattern 1: A computer cannot be the sole inventive step. This seems to be a direct result of the Alice decision. A software/algorithm patent that simply calls for automating a well understood, routine, or conventional activity will run into problems. This will probably also wind up including software that automates decisions or simply organizes information using mathematical processes.

In Comcast IP Holdings, LLC vs Sprint Communications Company LP, a Delaware judge rejected Comcast's patent claims because it was written so broadly that it couldn't be limited to a specific application. In fact, the judge went on to argue (convincingly) that as written the claim was simply trying to patent decisions that would be routinely made by operators.

The idea of replacing a telephone call was a theme through several of these decisions. In Eclipse IP LLC vs. McKinley Equipment Corporation a California court was faced with a case where the patent claim protected computer software that would ask a person to perform a task and then wait for the person to complete the task. One of the reasons that the court concluded that this was too broad for patent protection was that this transaction was simply replacing a phone call.

In a New York case, Dietgoals Innovations LLC vs Bravo Media LLC, found that the patent claims protecting menu planning software recited steps that could "be performed in the human mind, or by a human using a pen and paper" and that "a method that can be performed by human thought alone is merely an abstract idea and is not patent-eligible …"

Pattern 2: How the claims are written are important. In Planet Bingo vs VKGS LLC, Planet Bingo argued that "literally thousands, if not millions of preselected Bingo numbers are handled by the claimed computer program." The federal appeals court did not dispute this or even disagree that a large number of transactions could be an argument in favor of patentability. Instead they pointed out that these transactions were not included as part of Planet Bingo's patent claims, which only required "two sets of bingo numbers," "a player," and "a manager." The judge in the Eclipse IP LLC vs. McKinley Equipment Corporation case points out that in cases of complex software, the software itself may have to be recited in an actual claim.

Pattern 3: The courts may look kindly on patents with specific structures. In Digitech Image Technologies LLC vs Electronics for Imaging Inc et al, Digitech was trying to enforce a patent that coordinated and synchronized colors across digital platforms. The the appeals court rejected the patent claim as an abstract mathematical manipulation but they also addressed Digitech's argument that their system was hardware and software operating within a digital image processing system. Because the "digital image processing system" limitation was not included in the claims, the appeals court rejected the argument. However, the court did point out that if the software in the patent claim did exist in a physical or tangible form, described as a "concrete thing, consisting of parts, or of certain devices and combination of devices" then it would more likely be patentable.

Pattern 4: Back to the basics. There seems to be a general process and attitude within each of these rulings of moving back to the basics. In the end, they all took the Alice decision and then went back to the law as it was written and the prior case law to make their decisions. And while the logic varied, the judges seemed to converge towards the same conclusions.

At this time, I would suggest taking the above patterns with a large grain of salt. I have doubts that these cases are representative of the software cases in the system and I am suspicious about how quickly these cases were decided. It could be that these patents were so obviously bad that judgments could be made quickly and that more complicated and nuanced decisions are still in process will come along and gum up the works.