



The History of Software Patents

The U.S. Patent and Trademark Office historically has been reluctant to grant patents on inventions relating to computer software. In the 1970s, the P.T.O. avoided granting any patent if the invention utilized a calculation made by a computer. Their rationale was that patents could only be granted to processes, machines, articles of manufacture, and compositions of matter. Patents could not be granted to scientific truths or mathematical expressions of it. The P.T.O. viewed computer programs and inventions containing or relating to computer programs as mere mathematical algorithms, and not processes or machines. As such, software related inventions were considered non-statutory (see the BitLaw discussion on [patent requirements](#) for further information on the requirement that inventions be statutory).

In the 1980s, the Supreme Court forced the P.T.O. to change its position. The 1981 case of *Diamond v. Diehr* provided the first instance in which the U.S. Supreme Court ordered the P.T.O. to grant a patent on an invention even though computer software was utilized. In that case, the invention related to a method for determining how rubber should be heated in order to be best "cured." The invention utilized a computer to calculate and control the heating times for the rubber. However, the invention (as defined by the claims) included not only the computer program, but also included steps relating to heating rubber, and removing the rubber from the heat. The Supreme Court stated that in this case, the invention was not merely a mathematical algorithm, but was a process for molding rubber, and hence was patentable. This was true even though the only "novel" feature of this invention was the timing process controlled by the computer.

After 1981, the P.T.O. and inventors were left trying to determine when an invention was merely a mathematical algorithm, and when it was in fact a patentable invention that simply contained a mathematical algorithm. Although lower courts attempted to set forth this distinction in a clear manner, the resulting opinions were generally quite confused. What was clear was that the patentability of a software related invention depended heavily on the claims created by the patent attorney.

In the early 1990s, the Federal Circuit (the highest court for patent matters other than the Supreme Court) tried to clarify when a software related invention was patentable. The court stated that the invention as a whole should be examined. Is the invention in actuality only a mathematical algorithm, such as a computer program designed to convert binary-coded decimal numbers into binary numbers? If so, then the invention is unpatentable. However, if the invention utilizes the computer to manipulate numbers that represent concrete, real world values (such as a program that interprets electrocardiograph signals to predict arrhythmia or a program that analyzes seismic measurements), then the invention is a process relating to those real world concepts and is patentable.

In 1995, the P.T.O. decided it was time to develop guidelines for patent examiners that reflect these recent court decisions. After releasing draft versions of the guidelines for comment, the P.T.O. adopted guidelines for P.T.O. examiners to use to determine when a software related

invention is statutory and therefore patentable. These guidelines are analyzed in the next section.



Search: All

© 1996-2000 Daniel A. Tysver ([Beck & Tysver](#)) All Rights Reserved.
IMPORTANT: Please review the [legal disclaimer and feedback page](#)