

# SOFTWARE, INTERNET, AND BUSINESS METHOD PATENTS

Most people are familiar with patents for mechanical devices, but know less about the idiosyncrasies and rules for obtaining patents for software and business processes. This paper describes some of the unique attributes and requirements for software and business process patents in both the US and internationally.

## **What Can and Cannot be Patented in the US**

There are four categories of new discoveries that do not qualify for a patent because granting one would give an inventor an unfair monopoly on something that is “natural”:

- Mathematical formulas;
- Laws of nature;
- Purely theoretical phenomena (gravity, etc.); and
- Newly discovered substances that occur naturally.

In the early years of the computer industry, software was treated by the US Patent and Trademark Office (USPTO) as being a collection of mathematical algorithms, and therefore unpatentable. In 1981 the US Supreme Court ruled that an algorithm may be patentable if it works in connection with a specific apparatus. This opened the way for inventors to apply for software patents. However, the USPTO was opposed to software patents and took a narrow view in what it allowed. In 1998, a federal court (*State Street Bank versus Signal Financial Group*) overturned the USPTO and ruled that patent laws were intended to protect any method, whether or not it produced a “useful, concrete, or tangible result”. This dramatically expanded the applicability of patent law to software and business methods.

Technically speaking, software alone is not patentable. An invention that uses software code (or a series of process steps) to produce a useful result can be patented. There are specific ways in which software and business processes are described and claimed in a patent application. It is best to consult one of the references listed, to review patents in the same field in which you intend to apply, or to consult an experienced patent attorney to understand the ways to describe and claim a software product or business method.

## **Patents Versus Copyright and Trade Secrets**

Copyright is the traditional way in which software-based intellectual property has been protected. Copyright law is used to define who owns a specific piece of program code and has the right to develop and market this software or website. However, copyright protection is limited to the specific code or

web site. It still allows others to use the idea and implement it differently. Copyright is also difficult to apply to business methods. Trade secrecy (not providing source code for example) is another means to protect software.

Copyright should be used as a first line of defense. It is generally available (except for programming languages and business methods) and gives a basic protection. Patents give stronger protection, but aren't as easy to get. Trade secrecy also gives strong protection, but limits your ability to share and market an idea. What's the use of a great business method if you keep it a secret?

## **Utility Patents versus Design Patents**

Utility patents are what most people think of when they think of a patent. A utility patent is a patent for a useful functional implementation of a product. In contrast, a design patent gives a 14-year monopoly to a unique ornamental, aesthetic, or physical shape or design of a non-natural object--not its functional implementation. In some cases, one can get a design patent for an element of a computer program or a business method (a computer screen, for example). However, design patents are infrequently used to protect software and business methods.

## **Writing a Utility Patent Application**

The basic process for writing a patent application for a software product or business method is the same as for mechanical devices. The *Patent it Yourself* book and *Patent it Yourself* paper references present the basic approach. Some additional characteristics of software and business method patents are:

- The bulk of the “illustrations” for a software or business method patent are usually process flow diagrams. These have words in boxes, where most normal patent illustrations will have no words.
- If method claims are written, these claims will generally list all of the steps required to implement the novel idea embodied in the computer program or business method.
- It is also possible to write “machine claims” in which the entire system (including the software program or business method) is claimed.

## **International Software Patenting**

Patent protection is enshrined in the US Constitution, and it was a court interpretation that opened the way to software and business process patents in the U.S.

The situation in the rest of the world is different. It is almost always more difficult to get a software or business method patent abroad.

There is currently a significant debate in the Europe Union about software and business process patents. At of the time this paper is being written (May 2004), the situation is roughly as follows:

- The existing European Union law appears to restrict the granting of software and business method patents, however the European Patent Office (EPO), became significantly more lenient in interpreting this law (covertly since 1986 and overtly since 1998) and has issued over 30,000 software and business process patents.
- In September 2003, in response to protests by many members of the open source software community, the European Parliament voted for a directive proposal to stop the EPO from using a US approach to software patents.
- The European Commission (which generally supports software patents) and the European Parliament are now battling it out over what the future approach in the European Union will be.

I would argue that there is less need to get global software and business process patents protection than there is for mechanical devices because:

- The US is a bigger part of the world software economy than it is of other parts of the world economy.
- A US patent gives its owner the right to prevent others from using, manufacturing, or offering for sale a product incorporating the claimed invention in the US. Because it is difficult to create a software product and not offer it for sale in the US (putting it on the internet usually means people in the US could see and buy it), having a US patent effectively prevents others from exploiting this same software product.

### **Software Patents and Open Source**

The concept of open source software and patents do not sit well together. The philosophy of open source is that software should be treated like knowledge—it should be freely available and open to others to do with as they please, as long as the original author is given credit. This fits well with copyright law--the GNU license for open source is also known as “copy left.”

Patents grant the holder to a monopoly right to control how an invention is used. Patent licenses

often involve royalties. There have been several significant cases (the data compression algorithm used with GIF files, for example) where patent holders and the open source community have clashed.

### **Business Method Patents**

Many internet-related patents are business method patents. The 1998 court ruling also opened the way to business method patents. Some important examples include:

- The “one-click checkout” method used by Amazon.com.
- The algorithms used in the web marketing engine used by Overture. This is based on patents that were originally issued to a company called GoTo.com.

The process for obtaining a business method patent and the language used in the patent application are basically the same as for a software patent.

### **Conclusion**

This paper is an overview. Here are two in-depth references that I use and two papers I wrote recently to let you further explore this and related topics:

- Fishman, Stephan. *Web and Software Development, a Legal Guide*. Nolo Press. 3<sup>rd</sup> edition. ©1994-2002. <http://www.nolo.com>.
- Pressman, David. *Patent It Yourself*. Nolo Press. 9<sup>th</sup> edition. © 1985-2002. <http://www.nolo.com>.
- Vermeulen, Bert. *Patent It Yourself*. ©2004. See: <http://www.corp21.com/PatentItYourself.pdf>.
- Vermeulen, Bert. *Patent Strategy*. (c)2003-2004. See: <http://www.corp21.com/PatentStrategy.pdf>.

The US Patent and Trademark Office website at <http://www.uspto.gov> is also a good resource.

Please contact me if you would like more info. I recommend that you contact an experienced patent attorney or patent agent if you need specific advice.

### **About the Author**

Bert Vermeulen owns Corp21, a company that supports, incubates and advises businesses, entrepreneurs, and inventors around the world. For more information, see <http://www.corp21.com>.