

# Software Intellectual Property

Michael Matczynski  
Marcos Flores  
6.901/16.652  
December 9, 2004

## **Brief History**

The concept of software as a form of a stored program was first pioneered by John von Neumann who postulated that memory should be used to store instructions in numeric form, which would then be used to manipulate and control data.<sup>1</sup> This would eventually become known as machine code. Many of his ideas became the guiding light for much of today's electronic architecture and the way computers manipulate data. As time passed and technology progressed, machine code was made more understandable to programmers. Assemblers and compilers were developed to facilitate higher level languages such as FORTRAN, C++, and Java. With these higher level languages came the concept of operating systems, which is an infrastructure of modules that allows a higher-level language to interface with hardware. Application software became commonplace and popularity exploded with the introduction of the internet, first developed by ARPANET for collaboration between university scientists. Today, software continues to perform many important functions in all areas of industry, including accounting, manufacturing, banking, and telecommunications. Thus, it is absolutely critical that the United States of America have a well defined and effective system for encouraging software innovation.

## **US System**

There are three main ways to protect software intellectual property in the United States: trade secrets, patents, and copyrights. All three are generally used by a company

---

<sup>1</sup> <http://www.csupomona.edu/~hnriley/www/VonN.html>

to protect the research and development cost associated with producing a piece of software.

A trade secret is information that is not accessible to the public but which has economic value for its holder, which, if obtained by other entities, could cause the holder to lose his or her competitive edge<sup>2</sup>. Trade secrets are valuable because others do not know them. Restrictions exist on how trade secrets can be obtained and what information the holder can disclose. Trade secrets do not grant a monopoly to the holder, for if a company discovers a trade secret independently, that company is entitled to its use. This contrasts sharply with patents, which require an inventor to fully disclose an invention's workings to such detail that a person skilled in the art could produce the invention. A patent grants the holder the ability to block other companies from using the patented invention for any purpose.

The third way to protect software intellectual property is by obtaining a copyright on a piece of software. A copyright gives the owner the exclusive ability to control reproductions, derivatives, and distribution/display of work. The copyright covers the original expression of an idea. Whereas specific ideas can be patented if they are novel and useful, copyrights protect the original expression of an idea.

When protecting intellectual property, companies often obtain a combination of patents, trade secrets, and copyrights. With regards to computer software, many patent applications use the software's method or apparatus upon which software acts upon. Because source code is not required to be disclosed, the actual source code becomes a trade secret.

---

<sup>2</sup> Lundbergand Durant, pg 39

## **Systems in other Countries**

Intellectual property and software patents are handled in a variety of ways around the world. Some communist countries like China have weak intellectual property laws. On the other hand, world-leaders like the United States have a system in place to protect software development and encourage capitalism.

### ***Japan***

Japanese law does not specifically mention protection of computer software. However, the three requirements in analyzing the invention as a whole for patentability are:

1. If the application of a scientific principle involves the “physical or technical properties of an object” it is deemed patentable.
2. The invention must have industrial applications.
3. The invention must be novel such that those skilled in the art could not easily conceive of the claimed invention.

If these three criteria are met for computer software, then it will be deemed patentable by the Japanese patent office. This is important because Japan is a technologically progressive country and many companies invest billions of dollars per year for research and development of intellectual property, much having to do with software.

### ***Europe***

Many European nations have ratified and are signatories of the European Patent Convention, which created the European Patent Office (EPO). The EPO is a place where an individual can file a patent that can be valid in every one of the member nations, provided that it is examined and subsequently granted. Even with this in place, the

individual member nations' national patent laws can still trump the granting of a patent via the EPO if the national laws of that country dictate so. Therefore, several key Western European national patent laws will be examined in more depth in this section.

### ***United Kingdom***

The United Kingdom's patent law does not specifically mention the patenting of computer software, but generally, if a computer program carries out some process that is specifically linked to a physical apparatus or if the functioning of the computer program changes the way the actual computer operates than if it didn't include the program, then the filed patent could be considered patentable. Specifically, programs dealing with financial systems, document abstraction, or programs that only substitute for human mental steps without regards to a specific physical apparatus are not deemed patentable. As such, methods of doing business and mere abstractions of data would not be patented.

### ***France***

France's patent law adheres to article 52 of the European Patent Convention<sup>3</sup>, which states:

1. European patents shall be granted for any inventions which are susceptible of industrial applications, which are new and which involve an inventive step.
2. The following in particular shall not be regarded as inventions within the meaning of paragraph 1:
  - a. Discoveries, scientific theories and mathematical methods;
  - b. Aesthetic creations;

---

<sup>3</sup> European Patent Convention, Article 52.

- c. Schemes, rules, and methods for performing mental acts, playing games or doing business, and programs for computers;
  - d. Presentations of information.
3. The provisions of paragraph 2 shall exclude patentability of the subject matter or activities referred to in that provision only to the extent to which a European patent application or European patent relates to such subject matter or activities as such.

A process solely limited to a computer is probably not patentable in France.

However, if the process could work by other ways, other than using a computer, it would be patentable.

## **Germany**

In Germany, only technical processes can be patented. Special provisions have been added to handle software<sup>4</sup>. Because software is technical in nature, the following conditions apply to the granting of patents.

1. They concern the functioning of the computer so that they enable the direct cooperation of its elements.
2. They describe a certain structural form or improvement of a computer.
3. They enable a computer to be used for purposes different from those in the prior art.
4. The program enables a computer having known structure and elements to be used in a new and inventive manner.

---

<sup>4</sup> Lundberg and Durant, pg 398

5. Natural forces and the computer program or algorithm are related to each other so intimately that a technical result is achieved without the interposition of human mental activity.

The US is different in that software's technical nature does not itself alone enable patentability.

### ***India***

The patent law of India does not specifically mention computer programs but if a program is used for a process with industrial applications it could be patentable. This relatively weak and undeveloped patent system is surprising given India's recent boom in technological development. As Indian companies continue to grow and develop, the Indian Patent Office will need to adapt to ensure continued investment in computer software-related intellectual property.

### ***China***

The People's Republic of China's patent law considers software integrated with hardware to solve a technical problem that improve on prior art to be patentable. Like India's patent system China's is not very well developed. However, the expected future boom in China's computer industry would do well to provide a well-defined patent system. The current political situation in China will make it hard for an effective patent system to be in place. Recently, in 2001, China joined the World Trade Organization, and so they are currently in the process of improving and having well-defined patent system so as to attract investment in their country<sup>5</sup>.

---

<sup>5</sup> [http://www.nature.com/news/2004/041129/pf/nrd1595\\_pf.html](http://www.nature.com/news/2004/041129/pf/nrd1595_pf.html)

## Comparison of Patent Systems

In comparing how the United States patent system handles computer software-related patent claims, it is the most liberal out of every other country. Gerald Goldberg, the Group Director of Electrical Computer Group 2300 of the USPTO, which is the section primarily responsible for review computer software-related patents, said at an Annual PTO Meeting, “Do not claim a computer program...Do claim a computer implemented process.”<sup>6</sup> So, unless the computer program is a mathematical algorithm or a largely abstract theorem, it would not be considered patentable. Japan, on the other hand, grants patents to inventions and software that contains a specific applied technical solution. They are more concerned with programs that have industrial applicability than with mere abstractions or loosely defined processes. Concerning Europe in general, patents for software are issued so long as they have “a technical character or provide a technical contribution to the art and not simply a mathematical one.”<sup>7</sup> Many in the legal and software industry regard the method of reviewing patents in the United States as being a big cause of the high rate of growth in the recent decades of the software industry because companies know that their software code, applications, and processes will have a high chance of being protected.

## Brief Review of Patent Process in United States

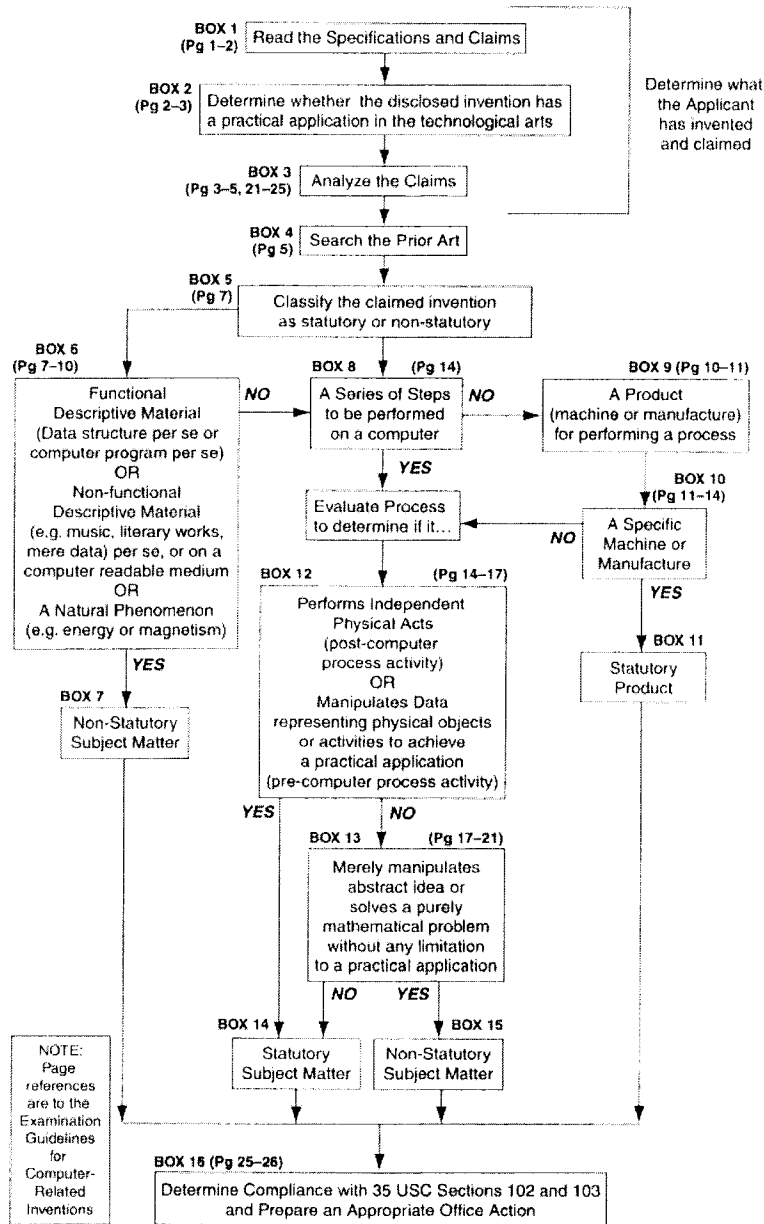
Figure 1 is a flowchart of the process that a patent examiner applies when deciding whether a software-related invention, especially one involving computer software is patentable.

---

<sup>6</sup> Durant and Lundberg, pg. 381

<sup>7</sup> Durant and Lundberg, pg. 382





**Figure 1: Process Flowchart for Examining Software Patents<sup>8</sup>**

The guidelines depicted above became effective March 29, 1996. One of the main points to address from the above figure are that when deciding if an invention is patentable, it is divided into one of three main categories:

<sup>8</sup> Durant and Lundberg, pg. 154

1. Functional or Non-Functional Descriptive material such as a data structure in the former or some artistic representation on computer readable medium, or a natural phenomenon.
2. A series of steps to be performed by a computer such as an algorithm, which is further subdivided into whether the software just performs some post-computer process activity or manipulates data to achieve a practical application, otherwise categorized as a pre-computer process activity
3. If the software is part of a machine or manufacturing process

As it is stated, the process for reviewing software-related patents goes through quite an extensive examination process to determine whether it is patentable.

## **Growth of Patents**

The number of software related patents issued by the United States Patent and Trademark Office has increased substantially during the past two decades. This is most evident in Figure 2.

## Software-Related Patents

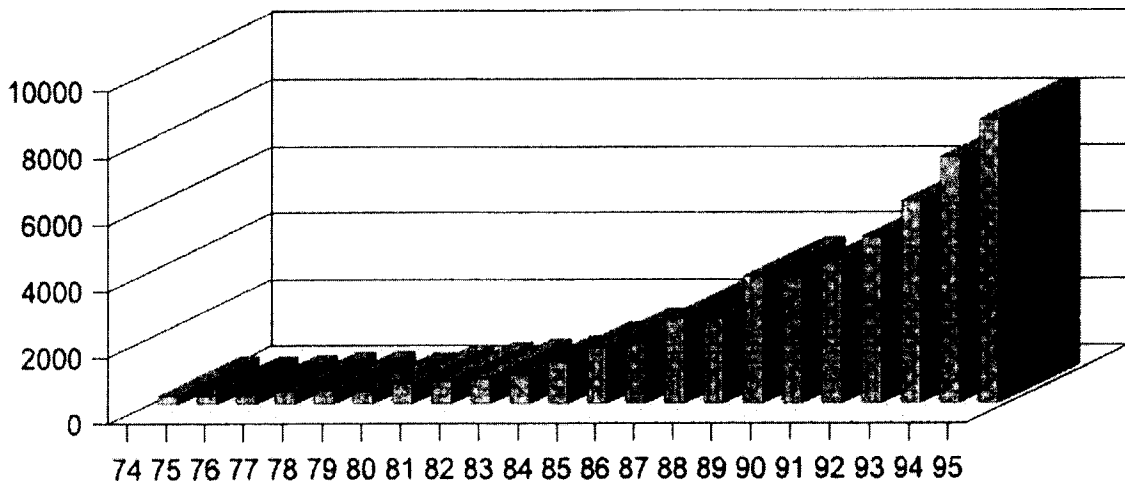


Figure 2: Growth of Software-Related Patents<sup>9</sup>

The table below shows the principal classes of software related patents. The two fastest growing areas of software patents are class 364 and 394 (see Table 1). In 1995, these two groups accounted for 7.16 percent of all patents issued by the United States Patent and Trademark Office. This is a substantial increase from 2.3 percent in 1980.<sup>10</sup> Recent estimates<sup>11</sup> suggest that the USPTO issues 20,000 software patents a year, with this number growing rapidly.

340	Communications Electrical
364	Electrical Computers and Data Processing Systems
371	Error Detection/Correction and Fault Detection/Recovery
395	Information Processing System Organization
701	Data Processing: Vehicles, Navigation, and Relative Location
702	Data Processing: Measuring, Calibrating, or Testing
704	Data Processing: Speech Signal Processing, Linguistics, Language
705	Data Processing: Financial, Business Practice, Management, or Cost/Price Determination
706	Data Processing: Artificial Intelligence

<sup>9</sup> Durant and Lundberg, pg.64

<sup>10</sup> Ibid.

<sup>11</sup> <http://www.researchoninnovation.org/softpat.pdf>, pg 1

707	Data Processing: Database and File Management, Data Structure, or Document Processing
711	Electrical Computers and Digital Processing Systems: Memory

**Table 1: Principal Classes of Software Related Patents**

American inventors tend to domain software patents. During the 1990's, 70% of all software patents went to inventors residing in the US. This contrasts with 53% of non-software related patents. Additionally, 70% of all software patents owned by companies went to American companies versus only 51% for non-software related patents.<sup>12</sup>

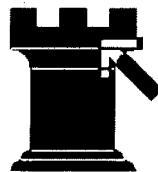
With such a substantial increase in the number of software related patents, searching for prior art has become more challenging. Because current lag times in the USPTO can be up to five years, other sources of patent news such as magazines and industry journals must be used.

## Notable Cases

With any system there are bound to be problems. Below are a few cases that reveal some of the shortcomings in current patent policy around the world, mainly in the area of software patents.

### **“Exclusive OR Patent”**

To demonstrate the ease of patent infringement, take the example of patent 4,197,590: *Method for*



Arrow drawn on rook using patented “exclusive or”

```
for (x = 0; x < height; x++)
  for (y = 0; y < width; y++)
    screen[x + xpos] [y + ypos] ^= arrow[x] [y];
```

**Figure 3: Patent 4,197,590: Method for dynamically viewing image elements stored in a random access memory array**

<sup>12</sup> <http://www.researchoninnovation.org/softpat.pdf>, pg 5

*dynamically viewing image elements stored in a random access memory array.* As Figure 3 shows, a mere 3 lines of computer code could cause patent infringement. Worse yet, code similar to this is written by programmers commonly because it is considered common knowledge.

### ***Amazon.com One-Click Patent***

One of the more famous software patent cases in recent years is Amazon.com's One-click checkout system, USP 5,960,411. This patent claimed a method for expedited checking out on an e-commerce web site by using information stored on a user's computer in the form of an Internet cookie. Shortly after Amazon.com began using this technology in 1999, competitor Barnes and Noble added a similar quick checkout mechanism to their own website. Amazon filed suit and reached an out of court settlement. This patent sparked an Internet controversy over software patents and a group was formed to encourage a boycott of Amazon.com. Critics claimed that Amazon's patent was simply a software implementation of an existing idea, being neither novel nor non-obvious. In an open letter from Amazon.com, CEO Jeff Bezos defended his company's patent and suggested ways to update the United States patent system. Bezos suggested "fewer patents, of higher average quality, with shorter lifetimes."<sup>13</sup> For the time being the firestorm has died down and Bezos hopes the patent system is updated before the next controversy.

### ***Microsoft Patents Double Click***

In 1997 Microsoft filed a patent for using short, long, or double-clicks to open different applications on a "limited resource computing device," such as PDA's. This

---

<sup>13</sup> <http://web.archive.org/web/20020213005852/www.amazon.com/exec/obidos/subst/misc/patents.html>

patent was issued in early 2004<sup>14</sup>. Critics have claimed that prior art exists and that this method is not novel. Virtually identical technology has been used on desktop computers for years. Because limited resource computing devices are often running the same version of the operating systems as desktop computers, opponents say the distinction made by Microsoft concerning “limited resource” devices is not valid.

### ***Australian Man Patents the Wheel***

In 2001 a man from Australia patented the invention of a “circular transportation facilitation device,” commonly referred to as a wheel<sup>15</sup>. Although the man has no intents to obtain injunctions against all wheel-utilizing companies, he said his case represents the problem with the Australian Patent Office. In recent years the Australian Patent Office has made patent applications easier to file and less expensive. This has substantially increased the number of frivolous patents. “All they're doing is putting a rubber stamp on it,” claims the inventor John Keogh. Even though this patent is not software related, it could be a predictor of times to come for the current United States patent system if policies are not changed. With the exponential increase in patents and no end in sight, the patent office will have to spend less time examining every patent, leading to cases like this in software patents.

### **Possible Improvements**

In order to suggest improvements for the way software patents should be handled in the United States, it is important to revisit the Constitution, which provides the basis for America’s entire patent system. Article I, Section 8 states:

---

<sup>14</sup> <http://www.newscientist.com/news/news.jsp?id=ns99995072>

<sup>15</sup> <http://edition.cnn.com/2001/WORLD/asiapcf/auspac/07/02/australia.wheel/>

“The Congress shall have power...To promote the progress of science and useful arts, by securing for limited times to authors and inventors the exclusive right to their respective writings and discoveries.”

Thus, a patent gives the owner the exclusive right to restrict others from producing their invention or idea for a limited time provided they disclose the workings of that invention to the public.

One of the major arguments against the way the USPTO currently examines software patents is that the exorbitant cost of obtaining a patent coupled with low amount of time and effort it takes to produce software code makes it especially hard for small to medium sized companies to excel as much as the larger companies in the software industry. Because smaller companies do not have the vast resources that large companies have, they are left behind and struggling to protect their work. What is at stake is meaningful and lasting innovation in the software industry that not only spurs its growth, but sparks more technological progress.

One possible change would be changing the length of time patents are granted. Currently in the United States the USPTO grants patents for 17 years. This may have been the optimal amount of time when most patentable inventions were large machines that took large amounts of research and development to produce. With software development research and development needed to create a patent is minimal. For example, there is a case known that a programmer made eight patentable inventions in the course of one day<sup>16</sup>. In order to prevent the patent system from holding back

---

<sup>16</sup> <http://www.okfn.org/wiki/IntellectualPropertyIssues>

innovation we recommend that software patents only be granted for a period of three years. Amazon.com founder and CEO Jeff Bezos agrees with us and says

“business method and software patents should have a **much** shorter lifespan than the current 17 years -- I would propose 3 to 5 years. This isn't like drug companies, which need long patent windows because of clinical testing, or like complicated physical processes, where you might have to tool up and build factories. Especially in the age of the Internet, a good software innovation can catch a lot of wind in 3 or 5 years.”<sup>17</sup>

Without software patents becoming so numerous, patent searches are becoming much more challenging to accurately perform. Even with a thorough patent search costing up to a million dollars, a software developer still isn't guaranteed that the software he or she produced is original. By shortening the lifetime of a patent, more trivial patents will be eliminated from the system and only inventions worth patenting will be pursued. This will greatly help reduce the amount of resources need to perform prior art searches.

Another possible change would be eliminating patents entirely and relying solely on copyrights to protect intellectual property of patents. This has the advantage of allowing industry to develop technology without fear of copyright litigation. One disadvantage is companies may be more reluctant to spend large amount of money and research and development on developing truly groundbreaking technology. Despite this, a software inventor's work and effort would still be sufficiently protected with a copyright. This protection is adequate enough and still effective such that

One currently complaint of the United States Patent system is that inventions that most software engineers consider obvious are being patented. This may be due to the

---

<sup>17</sup> <http://web.archive.org/web/20020213005852/www.amazon.com/exec/obidos/subst/misc/patents.html>



USPTO using the same definition for “obvious” as it does for the rest of patents. In the field of software, obvious must be defined more accurately in relation to its field.

“Obvious” patents may also be getting through the system because the majority of the USPTO staff is not trained in computer science. With the recent budget issues new patent investigators specializing in computer software are not being hired because the USPTO cannot offer them a competitive salary.

The above changes, if instituted by Congress and the USPTO, should be made retroactive. This would reduce the damaging impact that process patents such as Amazon.com’s one click system have had on the e-commerce and software industry. Once these changes are made, the software industry would most likely flourish even more than before.

### ***Micro and Macroeconomic Impact***

The micro and macroeconomic impact of adopting these changes could be staggering. Although it is hard to predict all aspects of these changes, on the whole software development in the United States would increase, creating higher demand for highly skilled software engineers. Additionally, because the majority of revolutionary innovation comes from smaller companies, the United States would continue to dominate the world in encouraging entrepreneurship. Adopting these policies all companies would be on the same playing field. Large companies like Microsoft and IBM will no longer have the advantage of having an entire legal department focused on filing for patents and suing companies who infringe on their patents. Additionally, the use and effectiveness of defensive patents would decrease if patents were valid for shorter periods of time. One possible disadvantage is that companies may use trade secrets more, preventing others

from building off of their ideas. Although a valid concern, the speed at which software is evolving would offset this and losses due to decreased research and development spending would be made up by more small companies being formed.

## References

Bessen, James and Hunt, Robert M. *The Software Patent Experiment*. (Research on Innovation and Boston University, Federal Reserve Bank of Philadelphia).

March 16, 2004. Accessed 12/8/2004.

<http://www.researchoninnovation.org/softpat.pdf>

Bezos, Jeff. *An Open Letter from Jeff Bezos on the Subject of Patents*. (Amazon.com).

2002. Accessed 12/8/2004.

<http://web.archive.org/web/20020213005852/www.amazon.com/exec/obidos/subst/misc/patents.html>

Biever, Celeste. *Microsoft Gains Double-Clicking Patent*. (NewScientist.com). June 3,

2004. Accessed 12/8/2004.

<http://www.newscientist.com/news/news.jsp?id=ns99995072>

CNN.com *Man Seeks Square Deal, Patents Wheel*. (CNN.com). July 2, 2001.

Accessed 12/7/2004.

<http://edition.cnn.com/2001/WORLD/asiapcf/auspac/07/02/australia.wheel/>

Durant, Stephen and Lundberg, Steven. *Electronic and Software Patent* (Law and Practice). Washington, DC. 2000.

Howard, Ken. *Patent fights rumble in China*. (news@nature.com). December 1, 2004.

Accessed 12/8/2004.

[http://www.nature.com/news/2004/041129/pf/nrd1595\\_pf.html](http://www.nature.com/news/2004/041129/pf/nrd1595_pf.html)

Open Knowledge Foundation. *Intellectual Property Issues*. November 11, 2004.

Accessed 12/2/2004.

<http://www.okfn.org/wiki/IntellectualPropertyIssues>

Riley, Norton H. *The von Neumann Architecture of Computer Systems*. (California State Polytechnic University, Computer Science Department). September, 1987.

Accessed 12/8/2004.

<http://www.csupomona.edu/~hnriley/www/VonN.html>

Stobbs, Gregory A. *Software Patents*. New York, NY. 1995.