

Software Patents Panel

Cristina Cifuentes
Centre for Software Maintenance
School of Information Technology
University of Queensland
cristina@it.uq.edu.au

Abstract

Over 6000 patents were issued last year in the US for software related inventions. Patent protection is available for software developments in Australia, US and Europe in addition to copyright protection. However, little use of the system has been made in Australia.

We review the different types of intellectual property systems, and the use of the patent system in Australia in relation to protection of software-related inventions. Our panelists will discuss the meaning of the patent system, rights obtained, applicability to software, myths, and the process of applying for a patent in Australia.

1. Patents and other Intellectual Property systems

Different types of intellectual property systems have developed throughout the years to protect the product of human's intellect and to promote the innovation of ideas. The most common systems for protecting computer software are: copyright, patents, and trade marks. We briefly describe each one.

Copyright protects the *expression* of an idea in a tangible (material) form. The recent Copyright Treaty by the World Intellectual Property Organization [13] confirms that computer software is to be protected by copyright as a *literary work* throughout the world. Literary works, such as books and poems, have traditionally been expressed in a written form which is readable to humans. Computer software source code is clearly an expression of ideas in a written form. And by extension, object code is an expression of the same ideas in machine language, hence it is also considered a literary work and therefore attracts copyright protection.

Patents protect the underlying *ideas* of a product which has commercial value in industry. Computer software incorporates ideas in the form of algorithms and data structures.

These ideas can be protected by patents if proven to be innovative enough and of commercial value.

Trade marks protect the *name* that uniquely distinguishes a company or product from the rest. Trade marks are available in regards to computer software in software names and software company names.

2. Myths or not

Several myths have been created in relation to patents protecting software-related inventions. People who oppose this system, e.g. the League for Programming Freedom [8], are normally against particular decisions taken in the early days of software patenting (when the Patent Office had little experience in reviewing software-related inventions), the large period of protection granted for software that may not live that long, the increased cost in development of new software that licenses someone else's patent, the apparent monopoly that some developers get with some patents, and the fear that innovation in software will be harmed by patents. However, several computer scientists and lawyers argue in a different way, e.g. [5, 9, 3]. Our panelists will discuss some of these issues.

3. Panelists

Our three panelists are leading experts in the area of software patents in Australia.

John Swinson is a Senior Associate at the Brisbane law firm of Mallesons Stephen Jaques where he specialises in intellectual property and technology law. John graduated with a major in Computer Science from the University of Queensland in 1986, and an LLB with First Class Honours in 1988. He was awarded a University Medal in 1988. As a Fulbright scholar, John obtained an LLM from Harvard Law School. He practised as an attorney at Kenyon & Kenyon in New York for six years, specialising in software and computer law. He has represented both large and small US software companies and universities.

David Webber is the partner responsible for the computer science patent practice of Davies Collison Cave. He is an experienced patent attorney who holds a Bachelor of Electrical and Electronic Engineering Degree with Honours from the University of Adelaide. He majored in VLSI semiconductor chip design, and Computer Science. He is a Registered Patent Attorney and specialises in patent, design and copyright advice for electrical, electronic, computer and telecommunications products. He has given lectures in computer software protection and integrated circuit legislation for Masters of Laws, Technology Law and Intellectual Property Diploma courses at the University of Melbourne. He has also given lectures in patent law and computer software protection for the Bachelor of Laws course at Monash University, Melbourne. He is regularly invited to speak on issues related to computer software protection and contributes to a number of intellectual property journals.

David Herald is a Deputy Commissioner of Patents at the Australian Intellectual Property Organisation (AIPO), with qualifications in Physics. His responsibilities include the setting of patent examination practice, and the conduct of dispute resolution processes before the Commissioner. David is well acquainted with the legal issues surrounding the patentability of software in Australia.

4. Chair

Anne Fitzgerald is a solicitor. She received an LLB(Hons) from the University of Tasmania, an LLM from London University, and LLM from Columbia University, and is a JSD candidate at Columbia University. From 1991 to 1995 she was a lecturer at the University of Tasmania Law School, where she taught Intellectual Property Law. In 1995, she was appointed by the then Justice Minister to the Expert Advisory Group assisting the Copyright Law Review Committee in its review of Australia's copyright laws. In 1996, she was appointed by the Minister for Science and Technology to the Advisory Council on Industrial Property.

References

- [1] G. Ahronian. Internet patent news service. Internet mailing list.
- [2] Patent office guidelines, 1992.
- [3] K. Bowyer. *Ethics and Computing*, section 8.7, pages 265–274. IEEE Computer Society Press, Los Alamitos, California, 1996.
- [4] CCOM Pty Ltd v Jiejing Pty Ltd and ors. 28 IPR 481, 1994.
- [5] P. Heckel. Debunking the software patent myths. *Commun. ACM*, 35(6):121–140, 1992.
- [6] IBM v Commissioner of patents. 22 IPR 417, 1992.
- [7] Patents act 1990.
- [8] R. Stallman and S. Garfinkle. Against software patents. *Commun. ACM*, 35(1):17–22,121, 1992.
- [9] J. Swinson. Software patents in the United States. *Journal of Law and Information Science*, 4(1):116–141, 1993.
- [10] D. Webber. Patenting computer software. *Computers & Law*.
- [11] D. Webber. Patents and trade marks for hardware and software. In G. Hughes, editor, *Essays on Computer Law*, pages 102–108. 1990.
- [12] D. Webber. Software patents: a new era in Australia and the United States? 5 EIPR 181, 1993.
- [13] WIPO copyright treaty. <http://www.wipo.org/eng/diplconf/distrib/94dc.htm>, Dec. 1996.

Background to Software Patenting in Australia

Anne Fitzgerald
Law School
Columbia University
New York, NY 10025
afitzg@law.columbia.edu

For many years, up to the late 1980s, the Australian Patent Office (APO)¹ routinely rejected patent applications involving computer software; for example N V Phillips Gloeilampenfabrieken's Application in 1966 [11].

Since Australian practice was diverging from the more liberal practices which were emerging in other countries, particularly in the USA, in March 1986 the APO published a new set of guidelines: *Guidelines for Considering the Patentability of Computer Program Related Inventions*. These guidelines incorporated the test developed by the US courts, known as the two-part Freeman-Walter-Abele test [9], the application of which subsequently led to the APO granting a number of software-related patents [10].

The question of whether computer software-related inventions constitute a "manner of manufacture", as required by the Patents Act 1990 [7] and are therefore patentable subject matter, first came before an Australian court in 1992 in *IBM v Commissioner of Patents* (the IBM case) [6], a decision of Mr Justice Burchett of the Federal Court. The case involved a patent application for an improved method of visually representing curve images in computer graphics displays. The application had been rejected by the APO on the ground that it did not disclose a "manner of manufacture". However, Mr Justice Burchett held that a method of producing an improved curve image by a computer utilizing a mathematical formula, which also meets the criteria of novelty and non-obviousness, is a patentable invention [12].

Following the decision in *IBM*, in August 1992, the APO issued new guidelines for determining the patentability of software-related inventions. The new test simply poses the question: "Does the invention claimed involve the production of some commercially useful effect?" Examiners are directed not to object to applications on the basis of the earlier US authorities which were adopted in the 1986 APO guidelines.

Illustrations of the test are provided in the guidelines [2]:

- A method-type claim must define a method which, either directly or by clear implication, embodies the commercially useful effect.

- A claim to a mathematical algorithm per se is not patentable, because it does not produce a commercially useful effect. A commercially useful effect can only arise when the mathematical algorithm is implemented in some manner to produce a result.
- A claim to a mathematical algorithm when used in a computer is patentable so long as a commercially useful effect is produced.

The guidelines provide the following illustrations of commercially useful effects:

- an improved image, where the image is commercially useful; and
- an algorithm that results in more efficient operation of a computer.

The approach taken in the *IBM Case* and the 1992 Guidelines has since been confirmed by the decision of the Full Federal Court in *CCOM Pty Ltd and anor v Jiejing Pty Ltd and ors* [4]. In *CCOM*, the court held that a computer programmed to enable the retrieval and display of Chinese language characters for assembly in text to be used in word processing was a *manner of manufacture* and therefore a patentable invention.

The decisions in the *IBM* and *CCOM* cases and the new APO guidelines, by simply requiring that a claimed invention must produce a commercially useful result, appear to have opened a generous scope for the patenting of software inventions in Australia. Although these developments do not yet appear to have had a major impact on the number of software patents being granted in Australia – in contrast to the US position where corresponding court decisions have stimulated an exponential growth in software patents [1] – the question is whether the patent system will, or should, be more extensively used in relation to software in coming years.

References

- [1] G. Ahronian. Internet patent news service. Internet mailing list.
- [2] Patent office guidelines, 1992.
- [3] K. Bowyer. *Ethics and Computing*, section 8.7, pages 265–274. IEEE Computer Society Press, Los Alamitos, California, 1996.
- [4] *CCOM Pty Ltd v Jiejing Pty Ltd and ors*. 28 IPR 481, 1994.
- [5] P. Heckel. Debunking the software patent myths. *Commun. ACM*, 35(6):121–140, 1992.
- [6] *IBM v Commissioner of patents*. 22 IPR 417, 1992.
- [7] Patents act 1990.
- [8] R. Stallman and S. Garfinkle. Against software patents. *Commun. ACM*, 35(1):17–22, 121, 1992.
- [9] J. Swinson. Software patents in the United States. *Journal of Law and Information Science*, 4(1):116–141, 1993.

¹Now the Australian Industrial Property Organisation (AIPO). For details see the AIPO web site at <http://www.aipo.gov.au>.

- [10] D. Webber. Patenting computer software. *Computers & Law*.
- [11] D. Webber. Patents and trade marks for hardware and software. In G. Hughes, editor, *Essays on Computer Law*, pages 102–108. 1990.
- [12] D. Webber. Software patents: a new era in Australia and the United States? 5 EIPR 181, 1993.
- [13] WIPO copyright treaty. <http://www.wipo.org/eng/diplconf/distrib/94dc.htm>, Dec. 1996.

Software Patents

David B. Webber
Davies Collison Cave
1 Little Collins Street
Melbourne, Victoria 3000
dwebber@davies.com.au

Patents for software developments are now well established and have been granted in selected jurisdictions for a number of years. In February 1996, the US Patent Office issued new guidelines to extend the type of software patents which can be obtained, a development which was forced upon it largely by the US Court of Appeals for the Federal Circuit. Previously patent claims to software processes could be obtained, but the guidelines now stipulate that claims directed to a computer program or a data structure stored on a computer readable medium are allowable. Accordingly, a US patent can be obtained to directly cover a computer program stored on disc or CD-rom, without requiring the program to be used or stored in some specific apparatus to infringe the patent. Australia has had a liberal practice in patenting of software developments since a Federal Court decision in 1991. The test which the Australian Patent Office uses to determine if a development constitutes patentable subject matter is simply to ask whether the development "involves the production of some commercially useful effect". The test has resulted in Australian applications largely being examined on their merits, i.e. whether the invention claimed is sufficiently novel and inventive, and time and money is not wasted on arguments as to whether the subject matter claimed is capable of being protected by a patent. New Zealand has also adopted the Australian practice. The Japanese Patent Office has issued examination guidelines to follow the US practice. The European Patent Office has indicated it may eventually have to adopt a similar practice but is hampered by a specific exclusion in the European Patent Convention regarding the patenting of computer programs.

Patents for software developments have now expanded to an extent that they are regularly obtained for all manner of software inventions, including telecommunications protocols, encryption algorithms and even features of video games. Although computer programs have been in existence since the 1950s, software patents have only been available in the U.S. since 1981 and have only come to the fore in a number of jurisdictions in the last four years. This has come as somewhat of a surprise to sections of the IT industry which have never had to deal with the issue of software patents. The 30 year lag also means that Patent Offices have not amassed a collection of patent literature which would

normally trace the development of a technology. This gap in the material has resulted in the granting of patent claims which are not truly novel and inventive and which understandably has drawn criticism from the industry. The U.S. Patent Office in particular is working to address the quality of its examination procedures and the new guidelines give explicit instructions in this regard. The efforts of the Software Patent Institute of Michigan to compile a complete database of known art should also assist. A good example of what can occur if the examination process is not thorough is the saga which erupted following the granting of U.S. Patent No. 5,241,671, otherwise known as Compton's multimedia patent. The patent was ultimately reexamined at the insistence of the U.S. Commissioner. Before issuing the new guidelines the U.S. Commissioner of Patents conducted Hearings which were open to all parties to voice their opinions on the granting of software patents. The view which people tend to take on the issue is either based on ideology or the manner in which they develop programs. Some developers want to be able to take any part they wish of the work of others, but do not wish others to take any part of their work. Of course what applies to one must also apply to the other. Some people have argued that patents should be abolished in general but this tends to ignore commercial realities and the fact that entire industries, such as the pharmaceuticals industry, are built on and to some extent defined by patents. Others specifically argue that software patents should be abolished, but this relies on an argument that software is inherently different to any other technology, for example electronic circuits, and therefore merits a special exclusion. The fact that software patents were not available for 30 to 40 years is largely an historic anomaly, due to the attitudes of hardware manufacturers, the intransigence of Patent Offices, and the commercial importance of software during that time. The only argument which can be clearly supported is that granting of broad and invalid software patent claims should be abolished.

Many have taken advantage of the availability of software patents and are using them to maintain their position in the market or to obtain royalties in order to recoup their development costs. Their use is not restricted to large corporations, such as Microsoft and Novell. A good example of how a software patent can be used to level the playing field and protect a development was the successful litigation Stac Electronics brought against Microsoft when Microsoft included a data compression routine in MS-DOS Version 6 which infringed two of Stac's patents. Stac was awarded \$120 million in damages and the two parties ultimately entered into a commercial relationship.