Journal of Information Law & Technology

Software and Business Method Patents

Chris de Villiers
Partner, Spoor & Fisher
c.devilliers@spoor.com

and

Tumelo Tshaya
Candidate Attorney, Spoor & Fisher
T.Tshaya@spoor.com

Lex Informatica Conference May 2008
Pretoria, South Africa

This is a **refereed article** published on 08 January 2009.

de Villers, C. & Tshaya, T., 'Software and Business Methods Patents', 2008(2) *Journal of Information Law & Technology (JILT)*,
<http://go.warwick.ac.uk/jilt/2008_2/devilliersandtshaya>

Abstract

Computer software has traditionally been regarded as unpatentable and patent offices in the USA and Europe, amongst others, rejected software-related inventions as being unpatentable subject matter. Following key court decisions in these jurisdictions, it was established that such inventions are patentable in terms of current legislation, and recent developments have related to the boundaries of protection and specific requirements for patentability. More recently, it has been accepted in these jurisdictions that business method-related inventions are patentable, provided that they meet a similar test for 'technical character' or a 'useful, concrete and tangible result'. Other countries have adopted similar approaches. In South Africa, where patent legislation is similar to that in Europe and the UK, it can be assumed that a similar approach would be adopted by our courts, although there is not yet any case law in this area. There is consensus that the lack of clarity is undesirable. Contradictory approaches by different South African government agencies to the desirability of 'software patents' are another reason for urgent clarification of South African law. The debate between those who favour patenting of software and proponents of Free and Open Source Software remains unresolved, while established software companies move to establish themselves in this area. Pending clarification of the law, whether by amendment of the Patents Act or a court decision, it is recommended that software developers apply for patents for new software- or business method-related technology, particularly if planning to do business internationally.

Keywords

Computer program, software, business method patent, technical effect, copying, piracy, open source, open standards.

1. Introduction

Before we talk about the possibility of patenting software, we need to know what we mean by 'software'. The word 'software' is often used to mean music, video or other content which is made available for purchase or use, as opposed to computer software. However, I am referring to 'computer software' in the sense of a computer program, as referred to in South African patent and copyright legislation. The boundaries between the two types of 'software' are easy to blur, due to the fact that media (such as music or streaming video) is most commonly provided in the form of digital files having some characteristics in common with computer software, and may be played on computers or other software-based devices.

1.1 Some Definitions: What Is Computer Software?

Wikipedia defines a computer program rather concisely as ‘instructions for a computer’ <http://en.wikipedia.org/wiki/Computer_program> and refers to two main functional categories of software, being system software and application software. System software includes the operating system which couples the computer’s hardware with the application software, while application software includes utility programs that help users solve application problems, and includes word processors, spreadsheets, web browsers and many others.

Other concise definitions for a computer program are ‘a sequence of instructions that a computer can interpret and execute’ <http://dictionary.reference.com/browse/computer%20program>, or ‘an organised list of instructions that, when executed, causes the computer to behave in a predetermined manner’ <http://www.vbtutor.net/vbtutor.html>.

Without getting too technical, the following major categories of computer software are recognised:

- Machine language/object code that controls hardware
- System software – typically operating systems such as MS Windows, Linux, UNIX, Apple’s Mac OS, etc.
- Application software – for example, word processors, web browsers, e-mail programs, spreadsheets, media players, databases, games, etc.

Application software, which is what most people think of first when the word ‘software’ is mentioned, is commonly written using advanced programming software tools which ease the task of converting a desired function into code. High level human-readable code, whether produced in this way or written directly in a programming language, is known as source code and can be analysed by software programmers to understand the techniques used in the software. The source code must be compiled (converted to machine code) or interpreted to be run on a computer.

1.2 The Piracy Problem

As a generalization, if complex software is distributed without access to its source code, it is difficult to decompile it into a human readable form in which it can be analysed and understood. However, it can still be copied in many cases, or reverse-engineered. If the source code is available, it is easier still for a third party to reverse engineer the software. Various technical measures can be taken to restrict or prevent copying and unauthorised use of computer software, but such measures have proved insufficient or ineffective in practice and it has been estimated by the Business Software Alliance (BSA) that over a third of all packaged software installed on PCs worldwide in 2005 was pirated, leading to a total loss of $34 billion to software manufacturers <http://w3.bsa.org/globalstudy/upload/2005%20Piracy%20Study%20-

%20Official%20Version.pdf>. This figure is based on the retail value of unauthorised software in use, and is disputed on various grounds, including the likelihood that many people using unauthorised software would not use it if they had to pay for it. One view is that a more realistic figure would be one-tenth of the above amount, which is still around $3.4 billion (or R25 billion).

According to BSA, South Africa's average software piracy rate decreased by one percent in 2006, but that still represents at least R1.2bn annually in economic losses. The BSA has further conducted a study which illustrates that if SA were to reduce software piracy, a stronger Information Technology (IT) sector could be created in SA <http://w3.bsa.org/southafrica/press/newsreleases/PR-1-28-2008.cfm>.

2 Copyright v Patents

Computer software can be protected against the unauthorised activities of third parties by means of two important types of Intellectual Property (IP), being copyright and patents. The nature and extent of the protection offered by these two forms of IP is somewhat different.

Section 11B of the Copyright Act 1978 gives the copyright holder the right to control reproduction or copying of the program, including the making of an adaptation of the program (meaning a version of the program in a different language or code, or on a different medium). Dean (2006) states that section 11B only protects the final stage of the computer program, that is, a stage where the computer program becomes a set of instructions to be used in a computer. Dean further explains that "work" produced in earlier stages of the development of a computer program, for example, flow charts depicting algorithm and other writings made along the way, are capable of being protected as literary works in terms of section 6 of the Copyright Act. In both sections copyright is effectively dealing with piracy and outright copying (unauthorised reproduction) of software/algorithm, or where a link can be demonstrated between the original software/algorithm and an alleged adaptation. However, whether section 6 or section 11B is used for protection, copyright is not able to protect the concepts behind the program code, so that a person who studies the operation of a particular architectural software package/algorithm, and then writes new code to implement the same features as in the original software/the algorithm, but without actually copying the original code/algorithm, will not infringe copyright. Another practical shortcoming of copyright protection is the fact that copyright in most works does not have to be (and in most countries, cannot be) registered. As a result, it is relatively onerous to prove ownership of an alleged copyrighted work for enforcement purposes. These are some of the obvious reasons why patents are of interest to software companies.

3 Backgrounds to Patents

In terms of patent law, the State rewards the owner of an invention with a twenty year monopoly in exchange for disclosing the invention to the public, so that only the inventor (or someone acquiring the invention from the inventor legitimately) is allowed to exploit the invention and is thereby able to 'enjoy the whole profit and advantage' of the invention for that twenty year period.

Traditionally, patents for computer software related inventions and business methods (which are often closely related) were rejected by Patent Offices, either because of specific limitations in the definition of patentable subject matter, or due to assumptions about the nature of computer software. It was thought that computer programs were more like books or compilations of data ('literary works') than other technological subject matter, such as automobiles, machines and chemical compounds, and ought to be protected by copyright only. (Although copyright does provide some protection, as mentioned above, it does not protect the underlying concepts in software related and business method inventions, and patent protection is potentially much broader.)

However, as the computer age progressed, computers and computer software took on an increasingly central role in business and industry. Today, computers and computer software are at the forefront of technology.

Although patent legislation has been slow to adapt in many countries, there have been developments in the law in this area in a number of countries in terms of legislation, patent office policy and court decisions. By way of example, where the Indian Patent Act was previously silent on computer programmes, the Act as amended by Act No. 15 of April 4, 2005, s 3(k) now provides that a computer programme per se is not an invention. In the USA, the Federal Circuit case of *State Street Bank and Trust Company v. Signature Financial Group Inc*[1] found that computer software and business method inventions are subject to the same standard of patentability as any other patentable system or method if the software produces '…a useful, concrete and tangible result'. Recently, the UK Patent Office published a Practice Note after the decision in *Astron Clinica Limited and Others v. The Comptroller General of Patents, Designs and Trade Marks*[2]. The practice note clarifies the scope and nature of patentable subject matter with regard to computer software <http://www.ipo.gov.uk/patent/p-decisionmaking/p-law/p-law-notice/p-law-notice-subjectmatter-20080207.htm>. These developments are nearly always in the direction of relaxing restrictions on the patentability of inventions involving computer software and business methods.

[1] [1998] 149 F.3d 1368; 47 U.S.P.Q.2D 1596

[2] [2008] EWHC 85 (Pat) Case No: CH/2007/APP/0466

3.1 General Requirements for Patentability

Before turning specifically to the patenting of computer software-related inventions and methods of doing business, it is important to understand the requirements for patentability in general, irrespective of the subject matter of invention.

The two universal tests for patentability are that the invention must be **new** and **inventive**.

Section 25(5) of the current South African Patents Act, No. 57 of 1978, states that an invention is new if it does not form part of the state of the art. The ‘state of the art’ comprises anything which has been made available to the public anywhere in the world, in any way, before a first patent application is filed to protect that invention.

Two important points to note from this definition are, firstly, that in order to be patentable, the invention needs to be a global first. Secondly, it is not possible to file a patent application for an invention which has already been sold or disclosed to the public. Therefore, a prospective patentee who has already been selling their software for even a short time has missed the patent boat in almost all countries (except possibly the USA and Japan).

The South African Patents Act 1978, section 25(10) states that an invention will be considered inventive if it is not obvious to a person skilled in the art at the time the invention was made. This second requirement for patentability is often the more difficult to overcome, as it includes a subjective assessment through the eyes of a notional ‘person skilled in the art’. Other countries have equivalent requirements.

3.2 Patenting Computer Software in South Africa

In addition to the above two tests, the South African Patents Act specifically excludes certain subject matter from being patentable. Section 25(2) of the Patents Act states that:

> ‘Anything which consists of –
>
> (a) a discovery;
> (b) a scientific theory;
> (c) a mathematical model;
> (d) a literary, dramatic, musical or artistic work or any other aesthetic creation;
> (e) a scheme, rule or method for performing a mental act, playing a game or doing business;
> (f) a program for a computer; or
> (g) the presentation of information
>
> shall not be an invention for purposes of this Act.’

At first blush, it appears as if a program for a computer is not patentable in South Africa.

However, section 25(3) states that:

> ‘The provisions of subsection (2) shall prevent, only to the extent to which a patent or an application for a patent relates to that thing *as such*, anything from being treated as an invention for the purposes of this Act.’ [Emphasis supplied]

It is not entirely clear what the legislature meant in subsection 3 when it used the term ‘…as such…’

There is currently no case law in South Africa interpreting the meaning of this subsection, and this therefore remains a grey area in South African Patent Law. However, since the provision in subsection 3 qualifies the exclusion in subsection 2, it is reasonable to interpret it as softening what would otherwise appear to be a complete exclusion. Burrell (1999) states that the words “relate to that thing as such” should be interpreted as meaning “only to the extent that such an invention relates to the excluded subject matter”. Burrell further suggests that the Copyright Act and Patents Act do not provide adequate protection for computer programs and that South African courts should adopt the approach outlined in the “Guidelines for Examiners in the European Patent Office” <http://www.epo.org/patents/law/legal-texts/html/guiex/e/c_ii_6_3.htm>.

Fortunately, as will be seen later, there have been many cases on this subject matter in the USA and Europe, in particular, and substantial progress has been made in clarifying what similar or identical wording to that found in these sections means. Recent cases and patent office practice notes in the UK have also clarified the position there. The net result is that computer software is largely patentable in foreign jurisdictions and we can extrapolate to some extent from such jurisdictions to South Africa.

3.3 Patenting Computer Software in the USA

The traditional approach of the US Patent Office to computer software and business method patents was to oppose such patents and to reject patent applications which were directed towards this subject matter. The US patent legislation provides: ‘Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor…’. So-called ‘scientific truths’ or ‘mathematical expressions’ are excluded from patentability, and the US Patent Office took the position that computer software was in effect a mathematical algorithm, and therefore not patentable.

However, in 1981 the Supreme Court case of *Diamond v. Diehr*[3] established that aspects of software with practical application were in fact patentable, and the US Patent Office’s

[3] [1981] 450 US at 175

attitude towards this kind of invention started changing. In fact, between 1982 and 1995, the US Patent Office granted 381 patents containing the word 'software' in their titles.

Without going into detail of the US case laws (which is beyond the scope of this paper), it can be said that the situation changed dramatically in 1998 with the decision of the Court of Appeals for the Federal Circuit in *State Street Bank and Trust Company v. Signature Financial Group Inc*. The Court made it clear that computer software and business method inventions are subject to the same standard of patentability as any other patentable system or method if the software produces '…a useful, concrete and tangible result.'

This decision was subsequently ratified in the case of *AT&T Corp v. Excel Communications Inc*[4] before the same Court.

The US Patent Office responded quickly and abandoned its directions to its examiners to reject patents merely on the basis that their subject matter was related to computer software or methods of doing business <http://commons.wikimedia.org/wiki/Image:US_granted_software_patents.png>. The resulting race to patent computer software and methods of doing business has been astounding. To date, tens of thousands of software or software-related patents have been issued in the United States and thousands more are being issued every year.

The situation at present in the USA is that software is patentable if it produces '…a useful, concrete and tangible result.'

Although the above is of particular interest to applicants seeking patent protection for their software inventions in the USA, the situation in the USA is somewhat removed from that of South Africa, in that the USA never had an equivalent statute to section 25 of our Patents Act. In general, the great increase in patenting of software related inventions in the USA has put pressure on patent offices in other countries to clarify their own positions.

3.4 Patenting Computer Software in Europe

In contrast to the position in the USA, articles 52(2) and 52(3) of the European Patent Convention (EPC) 1977 have almost identical wording to section 25 of the South African Patents Act.

Therefore, the case law in Europe is very relevant to the South African situation and provides an indicator as to the way our courts will interpret section 25 of our Patents Act.

The situation in Europe developed in a similar fashion to that in the USA with patents for computer software and business methods initially being rejected, although in this case based on specific wording in the legislation.

[4] [1999] 172 F.3d 1352

Again, without going into the long history of the change in attitude to the allowability of computer software, the major turning point came in 1998 when the Boards of Appeal of the European Patent Office issued a decision in regard to an IBM patent application in *IBM Computer Program Product*[5].

The invention in question was directed towards an 'asynchronous resynchronisation of a commit procedure'. The Board found that '…computer software is not excluded from patentability if, when it is run on a computer, it produces a further technical effect which goes beyond the normal physical interactions between program software and program hardware' (*IBM Computer Program Product*). This decision was important for a number of reasons, not least in that it recognised the technological nature of computer software.

In the light of this decision and further decisions by its Boards of Appeal, the European Patent Office changed its approach to examining computer software related inventions. At present, if a software-related invention has 'technical character', a patent for the invention may be granted, subject to the normal tests of novelty and inventiveness (*IBM Computer Program Product*). A decision in *Auction Method/Hitachi*[6], EPO Technical Board of Appeal Decision also confirmed that a method which involves a 'technical means' is an invention within the meaning of Article 52 of EPC.

Thus in Europe a claim to a computer-related invention (e.g. a process which can be implemented on a computer) is potentially patentable if technical features are involved in its implementation. Non-technical features will not be taken into consideration in deciding whether or not an invention meets the test of inventiveness. Technical features regarded as trivial will also not assist in this regard. This is consistent with the 'problem and solution approach' that is usually applied by the European Patent Office when assessing inventive step, i.e. the invention must involve a technical solution to a technical problem.

3.5 Comparison between US and European Patent Law

It is generally agreed that it is easier to obtain patent protection for computer software and/or business method type inventions in the USA as opposed to Europe, as the US criterion of a 'useful, concrete and tangible result' is easier to meet than the corresponding European 'technical effect' or 'technical character' requirement.

An invention consisting of software that controls a machine, for example, would most likely meet the technical effect requirement in Europe and would also certainly be patentable subject matter in the USA.

It would also appear that 'pure' business methods, without any computer software aspect, would not be patentable in Europe or the USA. In Europe, a method only involving economic concepts and practices of doing business would not be patentable. The European

[5] [1998] T1173/97 3.5.1 EPO Technical Board of Appeal Decision
[6] [2004] T0258/03 3.5.1

Patent Office judges technical character in business method inventions with reference to the following issues:

- what is the technical (as opposed to the commercial) problem solved?
- what are the technical considerations involved in the solution offered?
- how does the claimed subject matter affect the technical functioning of the method/system claimed?

A business method which is implemented using computer software, such as a new Internet business model, will be patentable if it has the required technical contribution to enable the business model to work.

In the USA, this kind of invention has in recent years certainly been easier to patent than in Europe. However, in the recent case of *In re Stephen W. Comiskey*[7], the US Court of Appeals for the Federal Circuit held that a pure business method consisting only of a mental process, without a computer system component, is not patentable <http://www.cafc.uscourts.gov/opinions/06-1286.pdf>. This brings the US and European approaches closer.

It should be noted that it is generally agreed that to merely automate a known human transaction process using well known automation techniques is not patentable. For example, to automate the filling in of a form would lack inventive content.

3.6 Other Countries around the World

The US and European positions represent the two major positions on patentability of computer software at the present time. The position in the UK is similar to that in Europe, although in recent times, following the UK Court of Appeal single judgment covering two cases in the *Aerotel* and *Macrossan* case (*Aerotel Ltd v. Telco Holding Ltd and Others, and Neal William Macrossan's application*), the UK courts and Patent Office were taking a relatively restrictive view on software and business method patentability compared to the European Patent Office. Specifically, the UK Patent Office has until recently had a policy of refusing patent applications with claims directed to a computer program, or to a computer program on a carrier <http://www.ipo.gov.uk/patent/p-decisionmaking/p-law/p-law-notice/p-law-notice-subjectmatter-20080207.htm>.

However, in the recent *Astron Clinica* case, *In re Astron Clinica*[8], the UK High Court held that if claims to a method performed by running a suitably programmed computer or to a computer programmed to carry out the method are allowable then, in principle, a claim to the program itself should also be allowable. The High Court stated that in these circumstances ‘the claim to the computer program must be drawn to reflect the features of the invention which would ensure the patentability of the method which the program is intended to carry out when it is run.’ Accordingly, the UK Intellectual Property Office has

[7] [2007] Fed. Cir. No. 2006-1286

[8] [2008] EWHC 85 (Pat); [2008] WLR (D) 12

issued an updated practice note, reflecting this thinking. This development aligns the UK Patent Office's approach more closely with US and European thinking.

Most other 'first world' countries such as Australia, Canada and Japan have followed a similar, but somewhat more conservative approach than the USA. In Australia, the current position is that the invention must be defined in a manner that is capable of technological implementation. A recent decision held that a patentable invention must have some physical aspect, which would not exclude software related inventions. New Zealand is in the process of amending its Patents Act 1953 with provisions based on the current Australian and UK Acts. The revised legislation does not exclude patentability of business method and software patents.

The position of the Japanese Patent Office (JPO) is similar to that of the EPO. The JPO was the first office to issue Examination Guidelines for examining computer software related inventions. The recent guidelines were issued in 1993, and they provided that computer software related invention is patentable on condition that 'information processing by software is concretely realised by using hardware resources'.

Article 25 (2) of the Patent Law of the People's Republic of China excluded 'rules and methods for mental activities' from patent protection. This article used to be an obstacle to the protection of computer software inventions, but China has moved from this viewpoint and presently provides that a computer software invention will be patentable if the computer software described in the invention application constitutes a technical program.

According to Section 3(k) of the Indian Patents Act, "A mathematical or business method or computer programmes *per se* or algorithms are not patentable". However, for computer software, the Indian Patent Office followed the practice of allowing a patent claim if the computer software was incorporated in hardware (embedded software) <http://www.mathewslaw.com/pub/pdf/paper-JPPS%202006.pdf>. This practice was given formal recognition by permitting 'technical application of computer program to industry' or 'a combination with hardware' as exceptions to non-patentability of computer programs.

As a practical matter, it is generally a good idea for prospective patentees to review the current position in the countries of interest when preparing to file a patent application for inventions of this kind. As a general rule of thumb, it can be said that in order to patent a software or business method invention in most countries, it is presently necessary for the invention to have 'technical character' more or less along the lines of the European requirements.

3.7 Where Does That Leave South Africa?

Until such time the question of patentability of computer software actually comes before a South African court, we do not have certainty on which position the South African courts will adopt.

What is very likely is that our courts will adopt an approach which will allow at least some kinds of computer software and/or business method inventions to be patentable, in line with the UK and Europe which have similar legislation.

Since obtaining a South African patent is relatively inexpensive compared with other countries in the world, it makes sense to go ahead and file a patent application in South Africa for an invention relating to computer software or a business method, rather than to sit back and wait for the position to be clarified. A patent application must be filed before an invention is made available to the public, so that failing to apply for a patent would simply lead to an irrevocable loss of potential patent rights.

Since patent applications are not examined for patentable content in South Africa, the South African Patent Office will, in any event, issue a granted South African patent for a computer software or business method related invention, and whether or not the patent is in fact valid will only be decided if the patent comes before the Commissioner of Patents in a legal action.

Recently, a number of articles have appeared in the media on the subject of so-called software patents, for example, in an article titled “SA minister slams software patents” <http://www.tectonic.co.za/?p=2304> and <http://www.aware.co.za/?q=node/66>. While some commentators are in favour of software being patentable, such as Paulo Ferreira, strategy manager at Microsoft South Africa (2008) < http://mybroadband.co.za/news/Software/3258.html> and others are vehemently opposed, for example, the South African minister of public service and administration, Geraldine Fraser-Moleketi <http://jonathancarter.co.za/2008/03/26/minister-slams-software-patents-microsoft-spreads-ignorance/ >, a common thread is the assumption that software is not patentable in South Africa. From what is said above, our view is that this is largely incorrect, and is based on an uncritical reading of section 25 (2) of the South African Patents Act, while ignoring section 25 (3) and the experience of other countries. Essentially, the two sections together could be interpreted as precluding the patenting of ‘pure software’ or program code, and ‘pure business methods’, as opposed to all software-related or business method inventions in general. (Such an interpretation would result in a neat dove-tailing of the Patents and Copyright Acts, with technological aspects of computer software being protectable by means of patents, and software code in a physical form, such as on a disc, being protected by copyright.)

It is by no means, as alleged by anti-software patent activists, that any software patent must necessarily be invalid. In fact, in the circumstances it could be argued that software companies would be foolish not to apply for patents until the position is clarified, either by a court decision or by amendment of the Patents Act. Most countries allow software-related inventions to be patented to a greater or lesser extent, and there is no reason to expect South African courts to take a different position. If software and business methods were held to be entirely unpatentable in South Africa we would be an exception to the world-wide trend, and there is no logical basis for assuming that our courts would interpret the existing legislation in such an extreme way.

It has been suggested that the South African Patent Office should examine patent applications involving software in compliance with the legislation. South African patent applications are not examined for novelty or non-obviousness of the inventions they cover. (This applies to all patent applications, not just software applications.) Similarly to the position in a number of other developing countries, the South African Patent Office does not have the resources to conduct a comprehensive technical and legal examination of patent applications. The onus is placed on a patent applicant to do the necessary homework to ensure that the patent is valid, and there is no doubt that many patents are granted, in all fields of technology, which are not valid. However, a patentee will not easily be able to enforce an invalid patent and may be penalised for trying to do so. For example, section 68 of the Patents Act provides that the Court can take the conduct of a patentee into account in a patent infringement case, where the patentee has attempted to enforce an invalid patent.

So, there is nothing ‘illegal’ or unlawful in filing patent applications for computer software or software-related inventions. Patents granted for such inventions in South Africa are likely to be valid if the patentee takes the necessary care to frame the patent claims correctly, bearing in mind the experience in other countries with similar legislation. Arguments about the desirability or otherwise of such patents tend to assume that software is a special category of subject matter that should be treated differently from other technology that can be patented. This debate continues to ebb and flow, with occasional flare-ups.

4. Patented Software v Free and Open-Source Software

A general discussion of software patentability cannot ignore the debate between the major developers of proprietary software and the advocates of free and open-source software.

At one end of the spectrum are a number of large software companies that have large portfolios of patents and see no reason why software should be treated differently from other patentable technology. At the other end are supporters of free software such as the Free Software Foundation, which strongly argues that software should be free. Occupying various positions in the middle ground are software developers who partly endorse the open-source approach, but want to keep their own improvements to open-source code proprietary, and some larger companies such as those that set up the Open Invention Network, which aims to create a shared portfolio of patents that the member companies can use against business opponents.

There has been extensive debate, particularly in Europe and the USA, on the issue of software patents and whether or not they are a benefit or a curse. Not surprisingly, companies that file software patents favour them while academics, open source proponents and smaller software developers without patents of their own are generally opposed to them.

An article in Fortune magazine <http://money.cnn.com/magazines/fortune/fortune_archive/2007/05/28/100033867/> mentions that Eben Moglen a software legal expert who advises the Free Software Foundation is of the view that “the free world says that software is the embodiment of knowledge about technology, which needs to be free in the same way that mathematics is free. Everybody is allowed to know as much of it as he wants, regardless of whether he can pay for it and everybody can contribute and everybody can share.’ Moglen holds the view that software is not patentable, on the basis that it is a mathematical algorithm.

Brad Smith, General Counsel of Microsoft, was once asked about the company’s recent decision to adopt a new licensing strategy, based on an analysis that showed that a large number of their software patents were being infringed by free and open-source software (FOSS). According to Microsoft, the Linux kernel itself infringes 42 Microsoft patents, while the Linux graphical user interfaces infringes another 65. The Open Office suite of programs, which is similar in functionality to Microsoft Office, allegedly infringes 45 more. It is also alleged that e-mail programs infringe 15 patents, and that a number of other FOSS programs infringe 68 patents.

The Free Software Foundation took on Microsoft and its allies in 2007 with amendments to their General Public License aimed at hampering a cross-licensing patent deal between Microsoft and Novell.

The Free Software Foundation developed its GNU General Public License (GPL), which has the largest usage of all open-source software (OSS) licenses, with the intention of preventing software vendors, who utilise open source software in developing their own software, from imposing restrictions on downstream users. The GPL requires that any software that is based on or derived from software distributed under that license also be distributed under the same license. Alternatively, if the software vendor obtains a patent based on software distributed under the GPL that patent has to be available for a free license for use by other vendors or must not be licensed at all. This restriction became known as the ‘liberty or death’ clause.

Microsoft’s deal with Novell avoids the terms of the GPL and effectively includes an agreement between the two companies not to sue one another’s customers, as well as involving patent cross-licensing and other terms.

One of the Microsoft/Novell deal’s term was that Microsoft could sell coupons that customers could trade in for Novell Linux subscriptions. Thus Microsoft was now seen as Linux distributor, hence subject to the terms of the GPL. Section 11 of the GPL was substantially rewritten in reaction to the agreement between Microsoft and Novell, and attempts to nullify the type of patent licensing deal in question. This is considered problematic by some commentators <http://www.aipla.org/Content/Microsites106/Electronic_and_Computer_Law2/Committee_Documents8/200718/2007_GNU_General_Public_License.doc>.

What is clear is that the conflict between supporters of proprietary, open-source and free software is ongoing, and the result is confusion and uncertainty for software users and developers.

4.1 Patented Software v Open Source Software in South Africa

The South African government is urging researchers to file patent applications to benefit the country's economy. The government has established incentive schemes, for example, the Patent Incentive Fund and Patent Support Fund, which are managed by the Innovation Fund. The Patent Incentive Fund's aim is to incentivise inventors who have filed and obtained granted patents in South Africa. While the aim of Patent Support Fund is to provide financial support for patent portfolios of inventions emanating from publicly funded research institutions.

The recent introduction of the Technology Innovation Agency Bill is stated to 'provide for the promotion of the development and exploitation in the public interest of discoveries, inventions, innovations and improvements, and for that purpose to establish the Technology Innovation Agency' <http://www.dst.gov.za/publications-policies/legislation/Technology%20Innovation%20Agency%20Bill.pdf>.

Furthermore, the government has also introduced an Intellectual Property Rights from Publicly Financed Research and Development Bill to facilitate with the creation of knowledge that is derived from public funding and to protect the knowledge in the form of IP rights, for example, patents, for IP that could have human wellbeing, military, economic and social benefits, <http://www.info.gov.za/view/DownloadFileAction?id=86995>.

So, on the one hand, government has a position of supporting the patenting of technology, including technology developed by publicly funded research and development. On the other hand, government's position regarding software seems to be aligned with the anti-software patent point of view, with a Free and Open Source Software policy which requires that all government departments and institutions make use of free and open source software, and migrate to an open software platform.

In March 2008, the Third Idlelo Conference on Free and Open Source Software and Digital Commons was held in Dakar, Senegal by the Free Software and Open Source Foundation for Africa (FOSSFA). South Africa's Minister for Public Service and Administration, Ms Geraldine Fraser Moleketi, stated: 'The adoption of open standards by governments is a critical factor in building interoperable information systems which are open, accessible, and fair and which reinforce democratic culture and good governance practices' <http://www.info.gov.za/speeches/2008/08032608451002.htm>. The Minister further said that patents are 'exclusive and anticompetitive in their nature' and that there is no reason to believe that society benefits from monopolies granted on computer program inventions.

It seems clear that there is some disunity within government and its agencies, when the Innovation Fund through its funding instruments is urging the filing of patent applications not only in the ICT sector to enhance economic growth and competitiveness, while on the other hand a government minister suggests that patenting of computer program inventions is undesirable.

The use of open standards for software interoperability is widely supported, not only by open source developers. However, government's definition of open standards is skewed heavily towards the notion that everything related to the standard should be free of cost and any intellectual property rights <http://www.info.gov.za/speeches/2008/08032608451002.htm/>. Such a view may be influenced by ideological considerations but it may also be based on the notion that open source software is 'free'. In the context of the GPL the word 'free' refers to freedom to use, edit and redistribute, rather than 'free' in the monetary sense <http://www.gnu.org> .

While some open source software (OSS) is in fact free, the use of much commercial OSS is not free in practice and is typically linked to a support subscription or other revenue generating model. The website <www.follars.com> has compiled a large amount of information on commercialisation of OSS, which is something of an eye-opener in this regard. An unknown author <http://it-online.co.za/content/view/73836/142/> quoted an IDC survey which found that worldwide revenue from standalone open source software reached $1.8 billion in 2006 and predicted that this revenue will reach $5.8 billion in 2011, representing a compound annual growth rate of 26% from 2006 to 2011. Another factor to be taken into account is the cost of migration from proprietary to open source platforms.

Companies like IBM, SAP, Sun, Intel, Hewlett-Packard and Silicon Graphics are committed to using open source software as a core part of their business and are investing significantly in this area. It is, ironically, questionable whether companies like these will not end up monopolising the open source field in time.

5. Where to from Here?

As has been said above, the extent to which software is patentable is evolving constantly and varies from country to country, and there is no certainty as to what the final position will be. There is a general consensus that the present lack of certainty regarding software patents is harmful and this need to be clarified as quickly as possible.

Particularly in the USA and Europe, various parties are lobbying for various different amendments to the legislation. Some want the scrapping of relevant restrictive sections in the legislation so that all kinds of business methods and computer software will always be patentable as long as they are new and inventive. At the other end of the scale, some parties are lobbying for a return to the traditional conservative position of the Patent Offices, or for amendments to patent legislation, to ensure that all patent applications for computer software and/or business method inventions are rejected out of hand.

Considering how far the various Patent Offices have moved away from this historical position, it is most unlikely that they will return to this position of their own accord.

In the USA, it may take a decision of the Supreme Court to clarify the parameters of software patentability finally. In Europe, what most parties want is a clarification of what is and what is not a 'technical effect'. Although (or perhaps because) there is a lot of case law on this question, there is still much uncertainty in this regard. At the moment, inventions have to be judged on a case by case basis to determine whether they would meet this requirement or not.

As there has yet to be a case in South Africa dealing with this subject, it seems that an amendment to the Patents Act, perhaps following developments elsewhere, will be required to provide clarity.

6. Conclusion

South African companies involved in producing new computer software must educate themselves about patents and what can and cannot be patented, both locally and in other countries. Companies should make an informed decision as to whether it is appropriate to file patent applications or not, considering their business models and their commercialization strategies. Due to the fact that patent applications filed at the South African Patent Office are automatically granted, and that the validity of a patent is only questioned if patent revocation proceedings are instituted in court, it is wise for a South African company to obtain as many patents in this area as possible. This can be looked at in business terms as a somewhat risky investment with possible long term returns.

Where a South African software company is interested in conducting business overseas, especially in the USA, patents are a must, whether to be used as a defensive shield or as part of a licensing scheme, for example. This applies to small and start-up businesses as well as larger companies.

One thing that everyone can agree on is that uncertainty on this issue is undesirable from almost all points of view. Rather than targeting software companies and making misleading statements suggesting that there is anything wrong in such companies trying to patent their technology, anti-software patent activists would do well to press for clarification of South Africa's legislation. They should appreciate, however, that there is also a strong counter view to theirs, to the effect that software-related inventions should be treated in the same way as other technology, and that South Africa's Patents Act should rather be amended to remove any outdated restrictions on the patentability of such inventions, as has been the trend in a number of other countries. Restrictions on the patentability of technology that was in its infancy 30 years ago, when the relevant legislation was drafted, may no longer be appropriate today. The South African government also needs to resolve its own position on software patents.

References

Burrell, T D (1999), *Burrell's South African Patent and Design Law*, (3rd Ed.), (Durban: Butterworths)

Dean, OH (2006), *Handbook of South African Copyright Law Service 13*, (Juta & Company)

Fraser-Moleketi, G (2008), Minister for Public Service and Administration, Free and Open Source Software and Digital Commons, Third Idlelo Conference. May 2008 http://www.info.gov.za/speeches/2008/08032608451002.htm

Mckay, D D and Krivoshk, D P (2006), 'Recent Developments in Indian Patent Law', 22nd Annual Joint Seminar Program, May 2006 http://www.mathewslaw.com/pub/pdf/paper-JPPS%202006.pdf

Parloff, R. (2007) Microsoft takes on the free world, *Fortune*, http://money.cnn.com/magazines/fortune/fortune_archive/2007/05/28/100033867/