



**15th BILETA Conference:
“ELECTRONIC DATASETS AND
ACCESS TO LEGAL
INFORMATION”.**

Friday 14th April 2000.
University of Warwick, Coventry, England.

Software Patents Pending?

Robin Widdison

Director

Centre for Law and Computing
University of Durham, UK

<http://www.dur.ac.uk/Law/deptment/widdison.html>

ABSTRACT

Recent twin decisions of the European Patent Office's Technical Board of Appeal have shifted its stance on software patents significantly. In this paper, I explore the background to this shift and suggest that the mindset to software patents enshrined in the Trade-Related Aspects of Intellectual Property Rights Agreement (TRIPS) is already being introduced ahead of formal amendment of the European Patent Convention and the national legislation that mirrors that Convention. In the light of the changing mindset, I go on to consider what the legal, economic and political may be. In conclusion, I discuss a longer-term scenario for global software protection.

KEYWORDS

Software protection - copyright - patents - TRIPS - *sui generis* protection

1. INTRODUCTION

1.1 Curiouser and Curiouser!

Software is like something out of Lewis Carroll's *Alice in Wonderland*. It is the most changeable and mercurial of substances. I use the term software here to refer simply to computer programs i.e. instructions that tell computer hardware what to do. I exclude for the purposes of this article data associated with the programs, physical media whether in the form of tapes or disks and peripheral information such as help files, user manuals and other documentation.^[1]

A substantial proportion of information can be digitised. It seems likely that soon, for example, any type of information that can be perceived - seen, heard, felt, smelt, or tasted - can be reduced to a collection of bits. Once in digital form, the information in question can be stored, processed and displayed by a computer. Furthermore, it can readily and rapidly be transmitted from one computer to any other computer regardless of distance, frontiers and physical obstacles.

From a digital point of view, there is absolutely no distinction between text, sounds, graphics,

photographs, music, animations, videos...and software. To include software in our list, however, may seem rather odd. Why is this so? Computer programs are, after all, only digitised information like the other examples. However, there is one vital difference between software and the rest. While text, sounds, graphics etc are generally passive in nature, software, by contrast, is essentially active. It is information with attitude! As Hart reminds us, *Computer programs are not only texts: they also behave.*^[2] Software is simultaneously both form and substance, both symbolic and functional. It both is and does. For this reason, software is conceptually very challenging.

Another point about the curious nature of software needs to be made here. It is a superb mimic. In particular, it has the capacity to make itself interchangeable with hardware. So, any given solution to a problem can often be implemented as either a combination of hardware and software, or as software alone. The significance of this characteristic will become apparent later.

Our understanding of the role and importance of software has changed dramatically over time. At first, we inhabited a hardware-orientated world. Computer programs were simply what you told your all-important hardware to do. Typically, it was written by the hardware manufacturer and arrived bundled with the hardware. Later on, software increasingly came to be seen as something separate from, albeit subordinate to, hardware. Such 'unbundled' software would be available from a software house that was independent of the hardware manufacturer. In the next stage, programs were perceived as not only separate to hardware but of equal importance to it. Today, we have reached the opposite pole to that from which we began. We live in a software-dominated world where hardware is nothing more than what you run your favourite software on.

1.2 Intellectual Property Rights

Intellectual property law comprises a collection of rights granted over the creations of the human mind. Commonly, these rights give the creator a limited monopoly over the use of his/her creation for a prescribed period of time. Such intellectual property rights can usefully be divided into two great families of regimes - *Two households, both alike in dignity*. Broadly speaking, the first family protects rights in aesthetic creations while the second family seeks to protect rights in industrial property.

Let us look more closely at the first family. The main social purpose of this type of protection is to encourage and reward original, creative works. Authors of aesthetic creations such as texts, musical compositions, dramatic scripts, works of art and films are all protected by copyright. In Europe, such protection typically lasts for a period of 70 years after the death of the author. Also protected through the copyright regime are producers of derivative works - essentially technologically advanced ways of embodying aesthetic creations. Derivative works include sound recordings, terrestrial or satellite broadcasts, and cable programmes. These works are generally protected for a period of fifty years from creation. A related right protects investment in the contents of databases for a period of up to fifteen years.

Turning now to the second family, the dominant social purpose here is to protect investment in the development of useful new technology, thus providing an incentive to finance research and development. So, for example, inventors of novel, industrial products and processes are protected by the patent regime for renewable periods up to a maximum of twenty years. So too, creators of industrial designs are protected by registered or unregistered design right for various periods. A third member of this family - albeit with a rather different social purpose - protects owners of registered trade marks whose intellectual property rights can be preserved indefinitely.

1.3 Which Family for Software?

If we have to determine which of the two families of intellectual property rights - copyright or industrial property - is more appropriate to protect software, we immediately hit a problem. As we have already seen, software is both symbolic and functional in nature. In symbolic, 'textual' mode, programs should surely be protected by copyright as aesthetic creations. However, in functional, 'behaving' mode, it would seem more appropriate to protect them as a species of industrial property. From a purely detached and theoretical point of view, it follows that software ought to be protected by both families. How, in fact, is it protected in Europe? We will look at question that next.

2. THE 'EITHER/OR' MINDSET

2.1 Protection by Copyright

When the issue of intellectual property protection for software first emerged as one of crucial importance, the dominant intellectual property mindset was exclusionary. New technologies were to be protected *either* by the copyright family *or* by the industrial property family.[3] It followed naturally from this stance that computer programs could be protected either as aesthetic creations or as industrial property, but not both. Which, then, was the more appropriate type of protection? With only two possible options, it is not surprising that there were two main schools of thought. The first school favoured protecting software as an aesthetic work by means of copyright. The second school, by contrast, preferred protecting software as a type of industrial property. From the outset, it seems to have been taken for granted by this second school that the right approach would be to protect software as a patentable invention.

Over time, the copyright family began to emerge as the preferred means of protection in Europe. Why should this be so? Here are five suggested reasons:[4]

* When clients represented software to lawyers, they tended to show it in symbolic rather than functional mode. This coloured lawyers' understanding of the concept of software.

* Copyright was already a mature, effective and widely-understood regime with a well-established jurisprudence.

* Generally, copyright protection required no registration. Thus, there would be no need to develop or adapt an examination system in order to subject software to demanding, high-threshold tests for registration.

* Copyright had long before become a conceptual dustbin for new technological forms of representation as they were created.[5]

* Copyright had developed a high degree of global harmonisation at an early stage. Software was clearly from the outset an international phenomenon. From this point of view, the two seemed made for each other.

On this basis, it appeared logical to classify source code - i.e. the symbolic representation of software in human-readable form - as an aesthetic creation. Object code - i.e. the symbolic representation of computer programs in machine-readable form - created more difficult conceptual problems. Insofar as object code could be seen as a translation of source code into another language, however, the fact that copyright readily protected translations from e.g. English into French seemed to provide a sufficiently close analogy to allow a machine-readable version of a computer program to be protected in the same way that the human-readable version.

If the European solution to the *either/or* conundrum was to be protection by means of copyright, it followed that the other leading contender - patent protection - should be excluded as a possibility.

This was effectively achieved at continental level by means of the European Patent Convention 1973 (The Convention) - a provision designed to harmonise patent law in much of Europe and to facilitate the acquisition of a bundle of European national patents by means of a single application. By virtue of article 52(2) of the Convention, computer programs were deemed not to be inventions and therefore unpatentable. As states that were signatories to the Convention were required to ensure that their domestic law was in step with the terms of the Convention, this exclusion percolated down into national laws.[6]

The rationale that was adopted for excluding software from the patent regime can perhaps be gleaned from the company that it keeps in the Convention and the Act. In both instances, computer programs are lumped together with 'schemes, rules and methods for performing mental acts, playing games or doing business' all of which are also deemed to be unpatentable. Why should this be so? The exclusions appear to be based on the notion that all the methods in question are abstract, intellectual constructs rather than 'down-to-earth' technical products or processes. As Gall argues:

The reason for the exclusion of programs for computers as such is that, like discoveries, scientific theories, mathematical methods and presentations of information, they are not of a technical nature. Patentability requires a specific technical application.[7]

While such abstract constructs might well qualify for copyright protection once set down and recorded in some way, they were deemed to be unpatentable. Interestingly, though, the exclusions from patentability that we have just looked at were not absolute. Article 52(3) provided a loophole. It declared that the exclusion from patentability applied only to software *as such*. So, for example, provided that a patent application was for something more, than - or different from - a computer program as such, the application would be considered in the normal way. Unfortunately, though, the key expression *as such* was not further explained in the Convention. So it was unclear, at first, how large the loophole was.

2.2 VICOM and the Aftermath

In due course, the loophole came to be considered by the Technical Board of Appeal of the European Patent Office (EPO) in VICOM's Application. [8] That case concerned hardware with a computer program which, together, would digitally process images in a sophisticated way. It was conceded by the applicant, however, that the hardware used could just as easily be a conventional computer. None-the-less, the Board stated:

Generally speaking, an invention which would be patentable in accordance with conventional patentability criteria should not be excluded from protection by the mere fact that for its implementation modern technical means in the form of a computer program are used. Decisive is what technical contribution in invention as defined in the claim when considered as a whole makes to the known art.[9]

Looked at holistically - i.e. hardware and software combined - the Board was of the view that the claimed invention escaped the *as such* exception and so was potentially patentable. There was a difficulty, however. In arriving at this result, the Board substituted one obscure concept - *technical contribution* - for another - *as such*. Subsequently, the European Patent Office (EPO) itself tried to clarify the meaning of this new concept in its official guidelines. [10] It gave the following examples:

(P)rogram controlled machines and program controlled manufacturing and control process should normally be regarded as a patentable subject matter.

The underlying thinking seemed to be that where software had an impact on the external world - e.g. an industrial machine or process controlled by a program - then the whole system, taken together could potentially be patentable. This seems both clear and logical - half an answer to the question of

the patentability of software. But what of the other half of the answer? The guidelines went on to say:

(W)here the claimed subject matter is concerned only with the program controlled internal working of a known computer, the subject matter could be patentable if it produced a technical effect.

This extract seems to be rather less helpful. Indeed, the continuing uncertainty over the patentability of software with an internal effect on a conventional computer has led to a prolonged paper chase through the decision making bodies of both the EPO and signatory states in search of a meaning for this new concept.^[11]

2.3 The IBM Twins

Now twin decisions of the EPO Technical Board of Appeal, both involving applications by IBM^[12] may have taken us a long way towards an answer to the second half of the question - at least within the framework of the Convention as it stands at present. In relation to software making a technical contribution to the internal workings of a computer, the Board has now made it clear in these cases what sort of technical effect it has in mind. Mere changes in the physical state of hardware by e.g. the manipulation of electrical currents are not enough. What is required is something else - *a further technical effect*. What does this third obscure concept mean? The Board goes on to give examples:

(A) patent may be granted not only in the case of an invention where a piece of software manages, by means of a computer, an industrial process or the working of a piece of machinery, but in every case where a program for a computer is the only means, or one of the necessary means, of obtaining a technical effect within the meaning specified above, where, for instance, a technical effect of that kind is achieved by the internal function of a computer itself under the influence of said program.^[13]

Where are we now? The overall effect of both VICOM and the IBM twin decisions taken together may well be to complete a *de facto* reversal the software exclusion enshrined in article 52 of the Convention. Tapper certainly thinks that this is so. He comments: *It is astonishing that the explicit provision of the Convention denying protection to computer programs have been read into virtual non-existence...*^[14] The original stance taken by the Convention seemed to be that software *as such* was presumed to be unpatentable unless the presumption was rebutted by evidence of something more. Now, it appears that the modern position is this. Software is presumed to be patentable unless that presumption is rebutted by evidence that the application is for software in isolation - a computer program *as such*. While this may appear to be no more than a change of emphasis, as any practising lawyer will know, it is an enormously important one. What is the basis for this assessment of the current situation? I believe that the EPO is now adopting an inclusionary, *and* mindset. I shall now examine why this should be so.

3. THE `AND' MINDSET

While Europe has wrestled with concepts like *as such*, *technical contribution* and *further technical effect*, the position in other key parts of the developed world evolved in a rather different way. In the United States, for example, the authorities seemed to have moved much more rapidly from the exclusionary, *either/or* mindset to the inclusionary, *and* mindset. As a result, software patents have been available much more readily in the United States than is in Europe for some time.^[15] This would be only of passing interest if it were not for the fact the United States is: (i) the world's most powerful economic giant; and (ii) the undisputed world leader in the development and deployment of computer technology and services.

It seems certain that the United States used its dominant position to have a decisive influence over key provisions of the Trade-Related Aspects of Intellectual Property Rights Agreement 1994

(TRIPS) - an agreement designed to form a basis for the global harmonisation of intellectual property laws. At first glance, however, TRIPS seems to be traditionally, exclusionary in the approach it adopts. The agreement expressly states the programs are copyrightable. Article 10(1) says:

Computer programs, whether in source or object code, shall be protected as literary works under the Berne Convention (1971). [16]

When we come to the section on patentability, though, Article 27(1) states:

Subject to the provisions of paragraphs 2 and 3, patents shall be available for any inventions, whether products or processes, in all fields of technology, provided that they are new, involve an inventive step and are capable of industrial application.

From our study of the European Patent Convention, we might well expect the abovementioned paragraphs 2 and 3 to contain an exclusion for software, amongst other things. However, when we read these paragraphs, we find that they only exclude immoral inventions, medical treatments and plant or animal varieties. There is no mention of software at all. Indeed the declaration in Article 27 (1) that patents shall be available for any inventions...in all fields of technology... clearly supports a view that the intention is that computer programs should have the same potential for patentability as any other product or process. In essence, TRIPS turns out to be the expression of a strongly inclusionary, and mindset.

This inclusionary TRIPS approach is now set to dictate the global nature of software protection, at least for many years to come. It is already having an enormous effect. In the twin IBM decisions, The EPO Technical Board of Appeal accepts with obvious regret that - as the EPO itself is not a signatory to TRIPS - it cannot unceremoniously dump the Convention software exclusion on the spot. The Board is, however, obviously straining at the leash to minimise the effect of the exclusion within the terms of the Convention against the day when the software exclusion is removed from the Convention in line with TRIPS expectations. Consider this comment by the Board:

(A)lthough TRIPS may not be applied directly to the EPC, the Board thinks it appropriate to take it into consideration, since it is aimed at setting common standards and principles concerning the availability, scope and use of trade-related intellectual property rights, and therefore of patents rights. Thus TRIPS gives a clear indication of current trends. [17]

Clearly, the Board itself was ultimately constrained by the terms of the Convention. However, pressure is now growing to bring the Convention into line with TRIPS by amending Article 52(2). [18] The most striking change, of course, will to take software out of the excluded category and place it firmly in the main stream of potentially patentable inventions. That is not all, however. The European patent has a younger sister - the Community patent. The Community patent was born in Luxembourg in 1975. [19] While the European Patent Convention enables inventors to obtain a bundle of national patents by means of one application, as we have seen, the Community patent regime, by contrast, is designed to provide a single, unified European Union-wide patent. However, since birth, the Community patent has been in a deep coma - it has never been implemented. Recently, though, there have been moves by the EU Commission to awaken the Community patent scheme. [20] If this does occur, it now looks certain that this regime too will treat software as potentially patentable in line with the TRIPS formula. [21] Is it only a matter of time, therefore, before the European approach on the protection of software is fully inclusionary - until both the copyright and patent regimes are freely available to authors or owners of new and inventive computer programs? If so, what are the implications of adopting the inclusionary, and mindset? We will examine this question in the next section.

4. THE IMPLICATIONS

4.1 Legal Implications

First, what are the legal implications of removing all bars on software patents? On the plus side, it will be wonderful to be free of the conceptual quagmire that currently surrounds European patents and their national equivalents. There will be no more need to wrestle with obscure concepts like *as such*, *technical contribution* and *further technical effect*. Another major change will be that the EPO will no longer have to live with a gross illogicality. As I mentioned earlier, one of the striking characteristics of software is that it can mimic hardware. As Hart has pointed out:

It is important to recognize that software and hardware are often interchangeable. To permit the grant of a patent for the hardware interpretation of a patentable invention but not for its software interpretation in my opinion borders on the ridiculous.[\[22\]](#)

In twin IBM decisions, the Board echoes these sentiments:

(T)he present Board finds it illogical to grant a patent for both a method and the apparatus adapted for carrying out the same method, but not for a computer program product, which comprises all the features enabling the implementation of the method and which, when loaded in a computer, is indeed able to carry out that method.[\[23\]](#)

It seems likely now that this anomaly will be swept away. Yet another benefit will be that the EPO will be cured of its schizoid stance of *prima facie* denying the patentability of software while, at the same time, actually granting many thousands of applications in respect of computer programs.[\[24\]](#)

From a legal perspective, though, it is not all good news. There is a risk that patent offices may be deluged with claims for software patents. However, it is not just quantity that is likely to put a strain on patent examiners. Given the rapidly increasing complexity of computer programs and the burgeoning state of the art, examiners may find themselves increasingly unable to cope adequately with qualitative issues such as novelty and inventive step as well. In practical terms, the system may simply be unable to cope.

4.2 Economic Implications

Let us now turn to consider some of the economic implications of the coming change. One of the key issues for intellectual property law is how to strike the optimum balance between, on the one hand, the rights of the creators/owners of existing property and, on the other hand, the interests the population in general, and would-be creators in particular. It is widely accepted that the public interest is best served by granting existing creators/owners a monopoly over e.g. their aesthetic works or inventions to encourage them to invest time, money and effort in creating, making and selling them. However, it is equally widely accepted that such monopolies must be so restricted in terms of extent, breadth and duration that the general population are not unfairly exploited and would-be creators are not stifled. How will the widespread availability of software patents effect this balance?

From one point of view, the effect may not be very great at all. Surely the software industry is much like any other industry? And anyway, removing restrictions on the potential patentability of computer programs is doing no more than taking away one fence from a lengthy steeplechase. Software, like all other inventions, will still have to jump all the other fences to complete the race. Applicants will have to establish, for example, that their software is novel, that it involves an

inventive step and that it has industrial application. As we have seen, though, Tapper is concerned that patent examiners may well be unable to cope adequately with the new demands placed on them. He expresses the fear that the result may well be a swing from automatic rejection of software patent applications to automatic acceptance.[25]

From another point of view, though, software patents may seriously distort and disrupt the existing balance between the proprietary rights of creators/owners and the interest that the general population has in the rapid development and ready availability of software. Bear in mind that, as we have seen, software is both a symbolic, aesthetic work and a functional invention. As the former, it is already protected fully by copyright. As the latter, it will soon be fully protected by the patent regime. As there is no proposal to do away with copyright protection, the result is that software will be doubly protected. Who will receive the benefits of this double protection? Despite arguments that the overhaul of the European patent regimes in general will remove a brake on the competitiveness of European companies,[26] as far as computer programs are concerned, most of the world's major software houses are now American. Surely, there must be real grounds to fear that, if European companies lost out in the past because of inadequate software protection, they may now be about to lose out because of too much software protection! As D'Amico reports:

The topic has also struck a nerve with Europeans software companies who fear that well-heeled American IT companies -- with a large cadre of patent lawyers -- will bully Europe's nascent software industry.[27]

Certainly, if we see an enormous land grab by the existing powerful software houses, followed up by well-financed, aggressive policies designed to defend those territories, it is hard to see how either the general population will benefit or would-be creators will be encouraged. On the contrary, there must be real cause for concern that tightening patent laws will stymie those trying to improve or create new computer programs, for fear of violating a patent.[28] Quite conceivably, such action by American software houses will result in a massive growth of the black market in computer programs, but this is hardly a satisfactory solution. Is it now too late to heed the warnings?

4.3 Political Implications

Finally, what of the wider political implications? As the Internet speeds up our move towards a global economy, the world-wide harmonisation of key aspects of trade law is no doubt both essential and unavoidable. Certainly, TRIPS was designed with that end in mind and any attempt to hold out against harmonisation in this or any other area of trade law would now seem positively antediluvian. However, accepting the inevitability of such harmonisation does not entail acquiescence in any particular regime indefinitely. Is a harmonised regime based on readier availability of software patents the best solution in the long run for Europe and for the world community as a whole? Or, is the truth of the matter that we are simply being subjected to American economic imperialism, as Tapper claims?[29] There certainly is a precedent for such an action. The US Semiconductor Chip Protection Act 1984 only granted protection to US nationals and nationals of such other states as offered equivalent protection. Not surprisingly, this had the effect of bouncing the European Union and its member states into adopting a US-style regime.[30]

5. CONCLUSION: A 'NEITHER/NOR' MINDSET?

Removing difficult and obscure concepts from the law is certainly desirable. Constructing a system that protects computer programs in both symbolic and functional mode is undoubtedly necessary. Achieving global harmonisation of software protection is absolutely crucial. Severe, long-term distortion and disruption of the balance between the rights of creators/owners of existing property, and the interests of the population in general and of would-be creators in particular could, however, prove disastrous.

Consider the possible reactions to such an imbalance. There is a spectrum of such scenarios. At the benign end, we might see a huge growth in communal software - programs developed and enhanced by many different programmers and offered cheaply, or completely free to users. An existing example is Linux the alternative to the Microsoft Windows family of operating systems.[31] Rather less benign, though, would be a massive expansion of the existing black market. In this scenario, legal protection of software is blatantly ignored. Technological protection is a challenge rather than an obstacle. Suppressive action against black market operators may, as William Gibson predicts in his writings, enhance the image of those operators so that they are increasingly seen as cyberspace guerrillas fighting in the cause of freedom. Driving black market activities into exile is hardly possible in the era of mass interconnectivity that is the Internet. Operators will simply relocate to unregulated parts of the world, and set up software havens. Then what? Economic sanctions, blockades, military strikes, or even software wars?

I believe that it is not a good idea to develop a global regime that overprotects software creators and owners. Is there an alternative? Certainly. It could involve a further shift beyond the current inclusionary, *and* mindset towards a *neither/nor* mindset. Computer programs, at least, would be protected *neither* by copyright law *nor* by patent law. Instead, we could renew our search for a tailor-made, *sui generis* regime. Such a regime might well have the following features:

- * Existing creators/owners of software would be entitled to register in order to obtain full legal protection of every aspect of their software - from initial design algorithms, through code listings to all functional characteristics.
- * Registration might be available to all creators/owners who could demonstrate that their software was substantially their own intellectual creation. No further account would be taken of such issues as novelty, inventiveness or industrial application.
- * The duration of legal protection would be appropriate to the useful life of a computer program - say a maximum of five years, renewable annually.
- * During the period of protection, other, would-be creators could decompile the software for the limited purpose of understanding the concepts, processes and techniques used.
- * During the period of protection, would-be creators could obtain a licence of right in respect of part or all of the protected software. The fee for such a licence would then be agreed by the parties or, in default, by a suitable independent body.

At present, it may seem that TRIPS is the very last word on global software protection. However, I believe that we will eventually see it as no more than the first word.[32] It is true that *sui generis* protection of computer programs is an unfashionable topic since the advent of TRIPS.[33] In time, however, growing distortions and disruptions in the global software market will force TRIPS signatory states to come together again in order to search for a software protection regime that is not only applicable world-wide, but also seen to be both even-handed and workable on that same global scale.

[1] See e.g. Millard C, 'Chapter 4: Copyright' in Reed C (editor), *Computer Law* (London: Blackstone, 3rd edition) at 112.

[2] Hart R, 'The Case for Patent Protection for Computer Program-Related Inventions' [1997] 13 *Computer Law & Security Report* 247.

[3] Demonstrated by the exclusion of aesthetic creations from patentability in article 52(2) of the European Patent Convention 1973 and mirrored in The Patent Act 1977 s 1(2).

[4] Compare Christie A, 'Designing Appropriate Protection for Computer Programs' [1994] 11 *European Intellectual Property Review* 486.

[5] However, Christie points out that software actually has very little in common with the other new technological forms of representation that have been shoe-horned into copyright protection.

[6] The United Kingdom Parliament enacted the Patent Act 1977 which mirrored the Convention. In particular s 1(2) of the Act reproduced the Convention's exclusion of software patents.

[7] Gunter Gall, Director of Legal Affairs at the European Patent Office, quoted in Hart R, 'The Case for Patent Protection for Computer Program-Related Inventions' [1997] 13 *Computer Law & Security Report* 247 at 250.

[8] [1987] 2 EPOR 74.

[9] VICOM's Application [1987] 2 EPOR 74 at 80-81.

[10] *European Patent Office Guidelines*, C-IV, 2.3 xii.1994.

[11] See e.g. Lloyd I, *Information Technology Law* (London: Butterworth, 1997) 280-289.

[12] Case T1173/97 and Case T0935/97. The latter has been reported in full at [1999] RPC 861.

[13] IBM's Application [1999] RPC 861 at 871.

[14] See comment to IBM's Application [1999] *Masons Computer Law Reports* at www.masons.co.uk/library/reports/0999/mclaw99d.pdf

[15] Lloyd I, *Information Technology Law* (London: Butterworth, 1997) 295.

[16] The Berne Convention for the Protection of Literary and Artistic Works 1886 (revised in 1971).

[17] IBM's Application [1999] RPC 861 at 868.

[18] See, for example, The EU *Commission's Green Paper on the Community patent and the patent system in Europe* (1997) 16-17.

[19] The Community Patent Convention 1975.

[20] See The EU *Commission's Green Paper on the Community patent and the patent system in Europe* (1997).

[21] *Patents: Commission outlines ambitious series of measures* (1999) at: europa.eu.int/comm/dg15/en/intprop/indprop/99/htm

[22] Hart R, 'The Case for Patent Protection for Computer Program-Related Inventions' [1997] 13 *Computer Law & Security Report* 247.

[23] IBM's Application [1999] RPC 861 at 874.

[24] Cohen L, 'The Patenting of Software' [1999] 16 *European Intellectual Property Review* 607.

[25] See comment on IBM's Application [1999] *Masons Computer Law Reports* at

www.masons.co.uk/library/reports/0999/mclaw99d.pdf

[26] For example, see The EU *Commission's Green Paper on the Community patent and the patent system in Europe* (1997) 1.

[27] D'Amico M, 'Europe considers loosening software-patent rules' (8 July 1999) *IDG News Service*.

[28] E.g. Stallman R, 'Saving Europe from Software Patents' in May 16, 1999 at <http://features.linuxtoday.com/stories/5960.html>

[29] A view he expresses in his comment on *IBM's Application* [1999] *Masons Computer Law Reports* at www.masons.co.uk/library/reports/0999/mclaw99d.pdf

[30] Lloyd I, *Information Technology Law* (London: Butterworth, 1997) 390-1.

[31] See Stallman R, 'Saving Europe from Software Patents' in May 16, 1999 at <http://features.linuxtoday.com/stories/5960.html>

[32] Compare the current debate over the protection of digitised information in general e.g. *Copyright and Related Rights in the Information Society - Proposal for a Directive* (1997) at http://europa.eu.int/comm/internal_market/en/intprop/intprop/1100.htm and *Amended Proposal For Directive On Copyright And Related Rights In The Information Society* (1999) at http://europa.eu.int/comm/internal_market/en/intprop/intprop/copy2.htm

[33] See e.g. Christie A, 'Designing Appropriate Protection for Computer Programs' [1994] 11 *European Intellectual Property Review* 486 at 487.