Quietly tying down Gulliver
The software patent fairy tale

The Battle of Trafalgar

One of the funniest things I saw this year was a photo of a rather large white cruiser bearing down on a small rowing boat on the lake outside the European Parliament. This was not the Battle of Trafalgar but the battle between the pro-software patent lobby (big white cruiser) and the anti-software patent lobby (small rowing boat). On this particular day, no blood was spilt, but this had been a little skirmish in a long running battle.

The battle has been played out for many years but in recent years has come to a head since the proposal of the Directive on the patentability of computer-implemented inventions (2002/0047/COD) (the '**CII Directive**') on 20 February 2002. What followed was a roller-coast ride through the EU legislative process culminating in the rejection of the CII Directive on 6 July 2005 where neither side could claim a complete victory.

Since its rejection, as far as the UK has been concerned, there have been four High Court decisions on the scope of patentability, a change in the UK Patent Office's ('**UKPO**') practice following one of these cases, (IN THE MATTER OF Patent Applications GB 0226884.3 and 0419317.3 by CFPH L.L.C. [2005] EWHC 1589 (Pat), ("**CFPH**") decided by Prescott QC) and a number of potentially important UKPO hearing officer decisions.[1] It has been a turbulent year. As will be discussed below, with CFPH we finally had some clarity in English law on the issue of software patents with a detailed judgment which explored the purpose behind the computer programs exclusion and sought to lay down a test for patentability firmly grounded on the European Patent Convention 1973 and public policy. However, it has been a brief calm in the storm because although this thorough approach was confirmed by the UKPO and the case *applied* by a High Court judge on 4th November (see IN THE MATTER OF UK Patent Application no.GB0108683.4 in the name of Cecil Lloyd Crawford (2005)("**Crawford**")), the waters have once again been slightly muddied by a decision of Mr Justice Pumfrey in the High Court which although heard several months before Crawford was only published on the 7 November (see, IN THE MATTER OF Patent Application GB 0017772.5 by Shopalotto.com Limited, [2005] EWHC 2416 (Pat) ("**Shopalotto**")).

The Shopalotto judgment should be treated as a pirate ship, which would be better at the bottom of the ocean, for it is weak judgment compared to CFPH and Crawford. The judgment lacks detail, ignores the careful analysis undertaken by Prescott QC (the case is not even referred to in the judgment) or the change in UKPO practice, fails to adopt a purposive approach to the EPC and to recognise the clear policy reasons for not allowing the foreclosure (i.e., monopolisation) of software under a patent system and gives weight to a body of decisions of the European Patent Office ("**EPO**") institutions that lack any legal or democratic foundation and which, against the purpose of the European Patent Convention 1973 ("**EPC**"), endorse the idea of the 'software invention', i.e., that a computer program can have patentable "technical effects" outside of any other claimed physical artefact or industrial process other than a computer.

The need for debate

There are a great many myths which have been propagated about the CII Directive and the desirability of software patents and there has been little *objective* debate about the CII Directive and, more particularly, about software patents *amongst IT lawyers.*

Why this article?

Put simply, the clouded and subjective debate over the CII Directive and software patents is detrimental both for IT lawyers and the IT industry.

Notwithstanding the recent English developments, there is little doubt that the debate over the patent system is to be had at a European level. Firstly, UK patent legislation to all intents and purposes transposes the EPC. Secondly, from an economic perspective, if we are to effectively compete against the US and Asia, we need to be able to guarantee innovation at a European level through pan-European laws.

With the Community Patent on the horizon, the issues debated on the CII Directive are bound to reappear and the outcome of such debate could have a long-term effect on the economic health and diversity of the UK and European IT sector. As IT lawyers, given our wealth is dependent on the health of such sector, we should be showing more of a passing interest. Furthermore, on the EU Commission's own figures from i2010, "*ICT account[s] for 40% of Europe's productivity growth and for 25% of EU GDP growth*". These are big figures by anyone's standards – the IT industry is vital to the European economy: at the heart of a successful Europe there has to be a successful home grown software industry

Given the above, it would seem absurd that a European directive would put at risk such an important part of our economy, but early this year, this is almost what happened. The fact that many people cannot imagine that those at the head of Europe may not be acting in our best interests is understandable when the debate has been shrouded in a campaign of disinformation. If not as IT lawyers, than as citizens whose economy is inextricably linked with the rest of Europe we should be concerned that we are not getting the full picture and take the moral responsibility to ensure that any debate is objective.

Getting religious

When you wish to conduct an independent survey about whether it is good to be catholic, if you are only going to ask the pope and his cardinals, you cannot call the results objective.

We have asked the pope and cardinals of the patent system and accepted their version of events not only on what the CII Directive intended to achieve but, somewhat more subtly, whether software patents are desirable. We, as lawyers, should recognise the conflict of interest but let me instead use the words of a more enlightened cardinal:

"*most patent lawyers -- most lawyers in general - …. unthinkingly spout pro-patent slogans. That is because most patent and IP and even other attorneys with an opinion on this issue mindlessly parrot the simpleminded economics with which they were propagandized in law school. Virtually every patent lawyer will reiterate the mantra that "we need patents to stimulate innovation," as if they have given deep and careful thought to this…..It does not take a genius to figure out why most patent lawyers are in favor of the patent system; and it is not because they have really studied the matter and dispassionately concluded that society is better off with a patent system -- it is because they don't want to see the system that pays the mortgage for them eroded or abolished.*"[2]

This article is not meant to be a sermon, what I have tried to do is independently review the issues at stake and leave it the reader to make his/her own analysis. However, if I am to preach about objectivity, then that must start with me the author. Do I believe in intellectual property rights ("**IPR**")? Yes I do and, as part of this, I see the benefits that patents can potentially bring in certain sectors. However, as one law professor recently commented, "*good policy does not just consist of 'more rights', it consists of maintaining a balance between the realm of property and the realm of the public domain*"[3].

I do not work nor have any financial interest in an open source software company nor am I getting paid, whether directly or indirectly, for writing this article or for expressing a particular view point. In fact, rather than standing to gain financially by encouraging the reader to adopt the views that I am espousing, I am more likely to have put a damper on any prospect of having a lucrative in-house position overlooking the M4 motorway (perhaps a good thing). If we are to see through the smog of disinformation, we first need to explore the key myths that have been put up in our way to obscure our vision

The CII Directive: the fairy tale

Myth 1: the European Competiveness Council, the European Commission and the Council of Ministers, as the white knights of European democracy and with our (European) best interests at heart, gallantly proposed a directive which, in its several drafts, sought to do nothing more than to codify and unify our existing laws on patents (commendable) in the area of computer related inventions whilst at the same time not extending the scope of patentability.

It is a wonderful fairy tale. Unfortunately for the knights, they were robbed by a bunch of European peasants, knocked to the ground once, got to their feet as nothing had happened (and didn't change their noble rhetoric) and proceeded to get knocked down for the second time, this time by more peasants and, if not a little belatedly, one or two noble men.

Those who are not interested in an objective debate about the benefit of software patents for the software industry want you to believe that the fairytale is true, and that MEPs for the last couple of years have been misled by a group of hairy open source hippies hell bent on preventing such unification of existing laws.

Upon what basis do they claim so valiantly that the CII Directive was not seeking to change the law (i.e., extend patentability) and that the hairy open source hippies, or peasants in my (or their) little fairytale, got it wrong? Well, to put it bluntly, because these bastions of European democracy told us that this was the case. In other words, the pronouncements of these institutions is rightly to be treated as divine against the heretical statements of the open source community who should be burnt at the stake for having the audacity to question the knights' intentions and disrupt the march of software patents across the world. The history of the passage of the CII Directive through the EU institutions exposes this myth.

If the actual words of the various drafts were so clear, why did the EU Parliament propose substantial amendments at the first reading, why did the Legal Affairs Committee of the European Parliament (JURI) vote overwhelmingly to restart the legislative process and, finally, why was a directive rejected for the first time in European history at its second reading? Was this purely the result of MEPs being misled by a misguided open source software community? To continue to make such a claim in the light of the history of the CII Directive is asking us to accept that MEPs are incapable of undertaking a simple analysis of the patent system or making independent judgements. It also discredits one of the greatest grass roots movements in recent European political history. European laws should be made for the benefit of Europe and its citizens. This may sound like common sense but for many arguing for software patents there seems to be an assumption that EU institutions should be working in the interests of the large IT corporations (the minority companies in the IT industry – see below). Laws should only be passed if they serve the public interest. For this to happen, or at least have a chance of happening, there needs to be a democratic and accountable legislature deciding these laws.

The Commission and Council of Ministers are not democratic; they are not elected by the public or directly accountable. The course of the CII Directive through the EU institutions is marked with several unsavoury incidents where the Commission and the Council of Ministers failed to give heed to the overwhelmingly clear wishes of the European Parliament, *the only* democratically elected EU institution. These failures, aside from the substantive debate, have left many question marks on the lack of democracy and transparency within the Commission and the Council of Ministers. As lawyers, why are we failing to comment on these shenanigans? Why was the Commission and Council of Ministers so keen to push through *their* draft – referred to ironically as the '**Common Position**” but which could never be described as having anything in "common" with the views of the EU Parliament - at all costs?

Had it not been for the undemocratic insistence of the EU Commission and Council of Ministers being so well documented by the anti-software patent lobby, coupled with the now notoriously unethical lobbying tactics of the pro-software patent lobby (reported to the EU anti-fraud commissioner by European lobby watchdogs[4] - I have never seen more dirty noblemen and such clean peasants), I and many others would have been none the wiser as to how far short the actual text of the proposed CII Directive fell from the boldly stated aims.

Part of the way Myth 1 has been spun has been dependent on the second myth of the legality of "software inventions" and the "existing *law*" which the Commission was attempting to unify.

"**Existing law**"

Myth 2: Patents for "software inventions" have been granted for many years by the EPO so they must be valid and part of the "existing law" which the Commission was trying to unify and codify.

When an act that is illegal gets carried out repeatedly without sanction it is understandable that many might subconsciously or otherwise consider such act legitimate. For our storytellers, the granting of patents for 'software inventions' has never been seen as prohibited but instead as 'part of the existing *law*'. Therefore, they could declare that Myth 1 was correct because for them, if the CII Directive allowed software patents it would not be an extension of patentability.

It is at this point that the seeds of objection to Myth 1 start to germinate, for the grant of software patents by the EPO *is not legal*, and therefore, is by definition not "existing law".

In CFPH, Prescott QC confirmed what many of us have been saying for some time, that current law has plenty of room for inventions which *include* computer programs – these are not prevented by the EPC. There are those that said we needed to desperately clarify the law and make the IT industry aware that they can patent inventions which involve computer programs and the presence of computer programs does not prevent them being patentable. If this had been the only outcome of the CII Directive, I would have been the first person lobbying for it to be adopted, for it would have done nothing more than confirm the spirit of the EPC. An invention which involves a computer program should not be precluded from patentability if it satisfies all other patentability criteria provided the computer program itself is not foreclosed.

However, the CII Directive was in substance going far further than this, it sought to legitimise the *undemocratic* grant of software patents by the EPO and the decisions of the Board of Appeals of the EPO (the '**BA**') favoured by the big name software players (the majority being non-European organisations) which would have resulted in the foreclosure of computer programs.

The EPO - the law of an administrative body

The EPO and the BA are administrative organisations, responsible for implementing the EPC not for amending it. The *source* of existing law is not the EPO and the BA but the EPC together with the interpretation of the EPC at a national level, interpreted by an independent judiciary whose livelihood is not dependent on the granting of more patents. As Prescott QC eloquently commented:

""*The EPO is not equipped with a staff of expert economists who are competent to decide if the patenting of business methods, or computer programs, would be good for our country and even if it was it would still be for our Parliament to decide*"[5]

Prescott QC's comments apply equally to all national patent offices including the UKPO. Patent offices are administrative bodies entrusted with the task of implementing democratically made laws. They are not legislatures. This is the way it should be. They do not have the democratic mandate, the independence or a team of suitably qualified economists to make law.

The EPO is a structure that lacks the fundamental characteristics of a competent legislature. It has no mechanism for conducting public consultations on crucial issues such as extending patentability to new areas, it is not democratically elected and its staff, as Prescott notes, are not suitably qualified to evaluate economic and public policy issues concerning patent laws. No detailed study has ever been conducted by the EPO as to the desirability of software patents and even if such a such a study had been undertaken it lacks the independence to properly evaluate any results yet, despite such constitutional issues, it has taken upon itself to extend patentability.

Given the lack of independence of both the EPO and national patent offices, we should be questioning the significant lobbying activities of patent offices across Europe and the EPO's own stance in the CII Directive debate.

Although, the BA may sound a ‘legal body’, it has no authority on substantive areas of law and is a technical appeals board. Like the EPO, its function is not to make new law but to implement the EPC. For the above reasons, as Prescott QC confirmed, the BA’s case law is not prescriptive, as it is a technical body not a legislature or judiciary – it cannot extend patentability by creating new laws (or not legally anyway).

The opposition to the Common Position did not come from those seeking to undermine an attempt to harmonise existing laws. There is a need to clarify the EPC in the area of inventions involving computer programs. However, what those campaigning against the Common Position saw was that the text of the CII Directive would not have resulted in the EPC being clarified (existing law) but instead would have succeeded in putting the case law of the BA, an administrative and technical body which had departed from the letter and spirit of the EPC, on a statutory footing and in so doing create *new law* and extend patentability; precisely the opposite of what the Commission and others claimed the text would do.

A story of divergence – the decisions of the EPO and BA

The EPC contains two important statements about computer programs:

1) they “shall not be regarded as inventions” (Article 52(2)(c)); and

2) that computer programs shall be excluded from patentability “only **to the extent** to which a European patent application or European patent relates to such subject-matter or activities **as such**” (Article 52(3), emphasis added).

There is no ambiguity as to whether computer programs are patentable under the EPC. They are “not inventions” and therefore cannot be patentable, or should that be cannot be patentable ‘as such’.

That’s right, in the run up to the CII Directive many MEPs were convinced by those supporting the CII Directive that the Common Position did not allow a US style system of software patents because computer programs were excluded from patentability “as such”. For most people this sounds satisfactory and many MEPs concluded there was not any intention to patent software and that the CII Directive would not have foreclosed software.

However, if you actually asked those who were pushing the Common Position, “can any software be patented?”, you would have got the response, “well, not software “as such”, but yes if the software produces a technical effect because then it is a software invention”. What is the difference between “pure” non-technical software and “technical software” or a “software invention”? The answer is, there isn’t any difference. At the end of the day what is being patented is the computer program except by the magic word “technical” what was previously not patentable becomes patentable. As Jim Warrant, board member of Autodesk (a world famous CAD software company) said in 1994 in his impassioned plea to the US Patent Office at hearings held over software patents, the concept of the “software invention” is…a pure invention:

“Thus, I respectfully object to the title for these hearings -- "Software-Related Inventions" -- since you are not primarily concerned with gadgets that are controlled by software. The title illustrates an inappropriate and seriously-misleading bias. In fact, in more than a quarter-century as a computer professional and observer and writer in this industry, I don't recall ever hearing or reading such a phrase -- except in the context of legalistic claims for monopoly, where the claimants were trying to twist the tradition of patenting devices in order to monopolize the execution of intellectual processes.”

Let’s go back to the fairytale. Imagine, that the king prescribed that knights were only allowed to graze sheep on his royal land and that cattle were out. Well, imagine if the knights turned round and argued that if the cow could sing and dance, it wasn’t really a cow but a technical cow which wasn’t a cow “as such” and could therefore graze with the sheep. Do we think the knights could pull the wool over the king’s eyes? After all, it may be a singing and dancing cow, but it is still a cow. Unfortunately, the fields are now full of cows. The fact that the pro-software patent lobby could make the dubious distinction between software “as such” and “technical software” as an excuse to foreclose software and thus fill the field with prohibited cows lies in the divergence of the BA and EPO from its administrative function of interpreting the EPC in a purposive way to extending patentability.

Starting with the decision in *Viacom* and culminating in the decision in *Hitachi,* the BA’s decisions and the practice of the EPO have resulted in software being effectively foreclosed by the bottomless pit and malleable concepts of, “technical contribution” and more recently “technical features”.

These are not phrases that appear in section 52 EPC (which defines what a patentable invention is)[6]. The concept of what is “technical”, and consequently “technology”, is not defined in the EPC. What the BA and EPO have in practice done is to interpret the word “technical” so broadly that “technical features” can be found in virtually all software[7]. As Prescott QC commented in CFPH:

“Now let me outline the practice of the European Patent Office. They look at the applicant’s claim, and ask themselves: does it have any “technical features”? If there are no “technical features” at all they reject the application, for not being an ‘invention’. <u>But they consider it is an invention if there is any “technical feature” at all. They take it very far.</u> Even paper, or ink, can count as a technical feature”.[8]

In doing so, they have failed to recognise that software cannot be considered as “technical” or “technology” for patent purposes, because, although “technology” is not defined positively in the EPC, it is defined negatively by reference to what is known as “excluded subject matter” (and computer programs are, as noted above, excluded subject matter). This is expressly recognised by Prescott QC in CFPH when he comments:

“*it will often be possible to take a short cut by asking “Is this a new and non-obvious advance in technology?” That is because there can often be universal agreement about what is ‘technology’….But sometimes it will not be possible without running the risk of error...If there is any doubt it will then be necessary to have recourse to the terms of Article 52 of the Convention*.” [9]

Unlike the EPO and the BA, Prescott QC uses the short-hand “technology” *with strict reference* to Article 52, which contains the list of exclusions including computer programs and recognises that “technical” and “technology, although useful short-hand, are ultimately limited by Article 52 EPC (in the context of determining patentability). The same cannot be said for Mr Justice Pumfrey’s approach in Shopalotto where he unashamedly grabs the loose reigns of the wild horse of “technical effect” unleashed by the EPO and BA and proceeds to gallop off with it:

“*Suppose a program written for a computer that enables an existing computer to process data in a new way and so produce a beneficial effect, such as increased speed, or more rapid display of information, or a new type of display of information. It is difficult to say that these are not technical effects, and, indeed, that the programmed computer, itself a machine that ex hypothesi has never existed before, is itself a technical article and so in principle the subject of patent protection. The real question is whether this is a relevant technical effect, or, more crudely, whether there is enough technical effect.*”[10]

Mr Justice Pumfrey, with respect, makes the same mistake as the EPO by endorsing the old “technical contribution” test (‘enough technical effect’) and succumbing to a broad definition of “technical” which fails to recognise that what may be considered as “technical” in common parlance does not mean it is “technical” for patent purposes (which requires a purposive reference to Article 52 EPC). In the context of a computer program, an increase in speed or faster display of information or new way of displaying information are not “technical” effects because they have no “physical” (technical) manifestation but result from the application of computer engineering, an exact science (rather than experimentation with changes to natural forces through the application of applied sciences). As the 17th senate of the German Federal Patent Court said in 2002:

"*If computer implementation of non-technical processes were attributed a technical character merely because they display different specific characteristics, such as needing less computing time or storage space, the consequence of this would be that any computer implementation would have to be deemed to be of technical character. This is because any distinct process will have distinct implementation characteristics, that allow it to either save computing time or save storage space. These properties are, at least in the present case, not based on a technical achievement but are pre-determined by the chosen non-technical method. If the completion of such a task could be a sufficient reason to attribute technical character to a computer implementation, then every implementation of a non-technical method would have to be patentable; this however would run against the conclusion of the Federal Court of Justice that the legal exclusion of computer programs from patentability does not allow us to adopt an approach which would make any teaching that is framed in computer-oriented instructions patentable.*"[11]

The use of "technical features" to justify the granting of patents cannot and should not be used without clearly referring back to the exclusions in Art 52 EPC. However, the broad brush approach of the BA and EPO to "technical features", and now followed by Mr Justice Pumfrey, is a long way away from the EPC exclusions and by allowing everything to be considered as "technical", the distinction between software "as such" and "technical software" or "software invention" mocks the purpose of the exclusion of computer programs in the EPC so as to effectively render it void. As the German Federal Court put it:

"*any attempt to attain the protection of mental achievements by means of extending the limits of the technical invention -- and thereby in fact giving up this concept -- leads onto a forbidden path. We must therefore insist that a pure rule of organisation and calculation, whose sole relation to the realm of technology consists in its usability for the normal operation of a known computer, does not deserve patent protection.*"[12]

A purposive approach

What is the correct meaning of the words "to the extent" and "as such"? Finally, law makers rather than technicians are starting to speak up, as Prescott QC states:

"*Article 52(3) indicates that a subject-matter is excluded 'only to the extent' that a patent relates to it 'as such'…. In the past it has led some people to think that you should be able to patent any new, non-obvious idea, so long as what is claimed as the invention does not consist only of excluded subject-matter. According to that reasoning you could patent an excluded item e.g. a computer program by the formal device of claiming some physical artefact (e.g. "A magnetic disk in which my program is stored", or "A computer when operating under the instructions of my program"). And indeed if it were just a question of interpreting Article 52(3) as if it were an Act of Parliament, they might have been right. However, it is not an Act of Parliament, and they were not right.*"[13]

This is a refreshing statement although we have had to wait a long time to hear it. The EPO, the BA and all the other sustainers of the "software invention" farce are "not right". You cannot and should not be allowed to subvert the purpose of an international convention which contains a clear exclusion on patenting computer programs by bestowing on the computer program some, physical, and hence non-abstract and tangible, character. For to do so would defeat the purpose of the EPC and as Prescott QC comments, the exclusions in Article 52 EPC need to be given a purposive interpretation[14] and in order to do this, we must understand the policy behind the exclusions.

Adopting a purposive approach has two key effects. Firstly, it makes it clear that you cannot have "software inventions" because by definition this can only ever be software, the invention can only consist of software and perhaps a computer or other physical artefact upon which the software is being run…at the end of the day this invention exists totally in the sphere of computers and software. However, it is submitted, that a purposive approach also allows one to satisfactorily deal with hybrid inventions (these are not "software inventions, but inventions which have a physical 'artefact' or 'process' (per Prescott QC) that is being claimed and which involve software).

"To the extent" and "as such" – dealing with hybrid inventions

"*The reason why computer programs, as such, are not allowed to be patented… ..[a]lthough it is hotly disputed now by some special interest groups, the truth is, or ought to be, well known…is because at the time the EPC was under consideration it was felt in the computer industry that such patents were not really needed , were too cumbersome (it was felt that searching the prior art would be a big problem), and would do more harm than good*[15]"

With these words, Prescott QC laid bare the argument of the pro-software patent lobby. Politicians have short term memories, as do most people who stand to make money out of software patents to the detriment of the software industry as a whole. The truth, as Prescott QC comments "*ought* to be known" but it has been conveniently forgotten by the pro-software patent lobby. Computer programs were excluded from patentability because patents in this area were seen as harmful to the IT industry and, this was not intended as a 'soft' exclusion but a 'hard' exclusion. As discussed later, the reasoning of our 1970's forefathers holds true today. Given such a clear statement of the public policy reasons behind the exclusion on computer programs it is, with the greatest professional respect, difficult to understand given that the "truth ought to be known" how Mr Justice Pumfrey could in Shopalotto state that " it [was] difficult to discern any underlying policy" for the computer program exclusion in the EPC. This failure to understand the policy reasons behind the exclusion of computer programs, leads him to conduct an analysis of patentability in a public policy vacuum resulting in a failure to adopt a purposive approach to the EPC and, inevitably (in such vacuum), to support the flawed view of the EPO and BA as to the words "to the extent" and "as such".

If, as Prescott QC does, we accept that the EPC excluded computer programs for strong policy reasons and that they should not be foreclosed, it is not consistent that the phrases "to the extent" and "as such" should infer that combining a computer program as part of a thing or process which is patentable should render the software *also part of the patent and therefore foreclosed*. A much more consistent interpretation of the effect of these phrases and the purpose of the computer program exclusion, is that Article 52(3) was drafted so as to deal with the 'problem' of preventing foreclosure of a computer program (against public policy) in the case of hybrid inventions which included a computer program. In order to do this, the most logical choice of words to delimit the excluded subject matter from the patentable subject matter would be to use the words "to the extent".

Therefore, in considering whether a thing or process is patentable which includes a computer program, the hybrid invention is foreclosed (assuming it meets the other patentability criteria) "*to the extent*" it does not involve the computer program as such. This means that the inventor gets the benefit of the patent for his hybrid invention, the presence of the software does not void the application but the software itself is not, at a formal level, patented (i.e., foreclosed). In this way, the ideas and functions of the software remain free as is necessary for public policy reasons (see discussion below) but the tangible, concrete and physical thing or process which has benefited from the application of a computer program is protected. Prescott QC implied this when he said:

"*the mere fact that a claimed artefact includes a computer program, or that a claimed process uses a computer program, does not establish, in and of itself, that the patent would foreclose the use of a computer program*"[16].

The presence of excluded subject matter should not prima facie prejudice a claimed artefact or process which uses the excluded subject matter but if the non-excluded subject matter is capable of being patentable, the excluded subject matter does not suddenly become part of the patent. "[T]o the extent" the now patentable invention includes a computer program, this element is excluded from the patent claims and hence is not foreclosed (hence Prescott QC does not talk of the 'claimed computer program' but instead of the claimed artefact or process because the computer program, being excluded subject matter, cannot be part of the claim).

The "Little Man" Test

Prescott QC, then goes on to talk about what is now been called the "little-man" test which was correctly, in my opinion, being interpreted by the hearing officer in the *Oracle* application[17].

"*There are many artefacts that operate under computer control.. and...many industrial processes that operate under computer control...The question to ask should be: is it (the artefact or process) new and non-obvious merely because there is a computer program? Or would it still be new and non-obvious in principle even if the same decisions and commands could somehow be taken and issued by a little man at a control panel, operating under the same rules? For if the answer to the latter question is 'Yes' it becomes apparent that the computer program is merely a tool, and the invention is not about computer programming at all.*"[18]

In the *Oracle* application, Oracle tried to apply for a patent of a mark-up language translation program by arguing that since what their program was implementing was a better way of applying the translation rules and that these rules could be applied by a little man, their invention was not to do with computer programming (and hence 'computer programs'). However, this application failed and although the hearing officer's reasoning has been criticised, it is submitted that the decision is not only consistent with the EPC but also Prescott QC's reasoning.

Prescott QC, it is submitted, did not intend the "little man test" to be a litmus test for whether something is excluded subject matter, i.e. if you pass the test you are not within the exclusion of computer programs. His words must be taken in the clear context in which they appeared. He was talking about hybrid inventions which may involve a computer program, i.e. unlike in *Oracle*, the little man test appears where the program is not the central element (this is the ***claimed artefact*** or ***process***), but is instead used to assist an examiner in considering the new and obviousness limb of Prescott QC's test. The test seeks to clarify that, in assessing newness and novelty, the presence of a computer program should not prima facie hinder an application if it appears to be the only "new" element, where it is clear that the function that the computer program is performing in the claimed artefact or process could be performed by a human being and that, the process if "humanised", is new and not-obvious having regarding to the state of the art. If the "little man" test is passed, it is the claimed artefact or process which is patented, not the computer program involved in the same.

However, the problem is that granting patents to hybrid inventions whilst at the same time preventing the foreclosure of software requires the EPO and BA to be able to:

- recognise the public policy behind excluding software programs in the first place;
- not conveniently forget history;
- not play a formalistic word game with "as such"; and
- not put their "customers" (mainly non-European customers when it comes to software patent applications successfully filed) first by taking upon themselves to extend patents to cover effectively software instead of its administrative function of upholding the terms of the convention as democratically adopted (including its spirit).

Since the CII Directive: (i) did not contain a clear definition of technology (recognised now not only by the anti-software patent lobby but also Prescott QC when he commented that the Common Position "*would have entrenched a test involving 'technical contribution' and 'technical features' that..[was]..too vague to be workable at the margin*"[19]); and (ii) relied on the "as such" definition, because of the EPO's and BA's track record on software patents, those arguing that the stated aims of the CII Directive had not been met knew that to have adopted the CII Directive in the form proposed by the Commission and others would have led to the legitimisation of software inventions, the patenting of pure software and the foreclosure of software in hybrid inventions resulting in an extension of patentability.

With the realisation that the CII Directive would not have achieved its commendable aims, those seeking to protect the majority of the IT industry had no choice but to try and modify the same to achieve the stated aims. After all, as Prescott QC comments, "*<u>[d]espite the prohibition</u> on granting patents for computer programs as such….the EPO has granted more than 40,000 of them*"[20]. If the EPO could grant 40,000 patents when, in law, they shouldn't be…imagine if they felt the BA's decisions were being put on a statutory footing.

A missed opportunity

The anti-software patent lobby wanted the directive to be a clarification of the EPC, not an extension. They sought to adopt a purposive interpretation based on an understanding that patents are not the divine right of inventors – public policy should dictate what is and is not patentable.

The pro-software patent lobby stated that their aim was also to clarify 'existing law' but as has been noted above, although the Common Position would have permitted hybrid inventions involving computer programs, pure software would have been patented and also all software in hybrid inventions would have been foreclosed, hence creating a result never foreseen by the EPC, against the purpose of the EPC, an extension of patentable subject matter and a move against public policy as it was then and as it is now.

What we needed, and I think we still need at a European level (although Prescott QC has got very close to doing this at the UK level) was to clarify how the EPC applied to hybrid inventions involving software, but to do this in a way which didn't rewrite the exclusion of computer programs and result in the foreclosure of software. Unfortunately, once the pro-software patent lobby realised that we were on the eve of achieving this, they did a u-turn and promptly proceeded to kill the directive they had lobbied so hard to achieve – the directive, was in the end not killed by the open source movement although this truth, which *ought* to be known has failed to be presented "as such".

Only the hippies

Myth 3 - The only interest group protesting about the CII Directive allowing software patents through the back door were open source software companies and individual developers.

This is a very important myth for the pro-software patent lobby. Often I have seen press releases talking about the "open source community" hampering the CII Directive without any mention of proprietary software companies also protesting about the CII Directive's substantive effects. The intended purpose of this distortion in reporting is clear, it seeks to polarise the debate as one between proprietary software companies (for software patents) and open source software companies (anti-software patents and 'haters' of intellectual property rights). The truth however is very different.

As discussed in the next section, software patents do not discriminate between the proprietary and open source sectors, the damaging effects on innovation are the same, it's just that in the open source sector the effects are compounded by the sheer scale of independent development which is inherent in large scale, worldwide, development projects. Software patents affect any individual or organisation writing software, regardless of licensing model, because they grant monopoly rights to the building blocks of software engineering and hence constitute a barrier to anyone writing software, but in particular to Small and Medium-Sized Enterprises ("**SMEs**").

The big porky

Myth 4 – Software patents are good for the IT industry and necessary for innovation

"*<u>Most economists have doubts whether economic efficiency, i.e. increased overall welfare, is achieved by having or making computer program related inventions patentable</u>. This caution is supported by the continuing, indeed growing, concern in the USA on the issues surrounding patents on computer program related* *inventions. The debate in the States is not finished.*"[21]

The above statement was not written by an open source software company. The statement was delivered as part of an independent report commissioned by the EU Commission prior to introducing the CII Directive. In no uncertain terms, the authors highlighted the conclusion that software patents were unlikely to produce benefit to our (European) economy and this was a report commissioned by the EU whose undertones were particularly conservative. However, within a couple of years, certain EU Commissioners would have forgotten all about this careful analysis (confirmed by many subsequent studies) and would be trying to convince us that software patents are beneficial to our economy.

IPR protect just about every element of a software product, including the lines of code, the graphics, the sounds, the front-end text, icons and the general look and feel. The missing piece of the jigsaw are the functions of the software, which would be protected (or monopolised) if software patents were allowed (whether subversively or otherwise). The pro-software patent lobby wants us to believe that software patents or "inventions" (as they like to call them) are a *natural* and acceptable progression of the patent system and that there is nothing wrong with foreclosing software through the patent system. What we are being told is that all these forms of IPR protection are not enough, we need stronger IPR in the forms of patents – developers should be given a monopoly over functionality and that this is vital for the continued competiveness of the European IT industry. Who is right?

Why have patents?

The purpose of patents is to stimulate innovation by rewarding those who make inventions with monopoly rights. However, patents have for decades been only granted (in accordance with a purposive approach to the law) for technical inventions, i.e. inventions which have a *physical* effect. This is often expressed in different ways in each member state but it is fair to say that, drawing from the main concepts, there is a convergence towards the idea that the invention must be "technical", have a "practical form" and "physical" effects, represented in some form of apparatus or device. Therefore, in CFPH Prescott QC litters his judgement with references to claimed "artefact" or claimed "process" and in discussing attempts by 'special interest groups' to subvert the purpose of Article 52 EPC he refers to them trying to claim the abstract computer program together with a "physical artefact", thus recognising that the patentable subject matter is that part which is the physical manifestation of the idea. What has historically been protected by patents is not ideas themselves, but "*a new way of putting an idea to work by using controllable forces of nature*."[22]

For such technical innovations, society has considered it right that, given such inventions often involve a significant amount of experimentation and provide benefits to us, we need to encourage inventors by giving them the economic incentive of a monopoly right in return for publication of the invention and the system used to do this is the patent system. For such technical inventions which create a change in natural forces, there is really no other suitable intellectual property right which rewards and protects the inventor's time and effort in undertaking such experimentation, which because we are in the realm of applied sciences rather than exact sciences (like mathematics) can be substantial.

However, ideas (mental thoughts, abstract or logical processes) have not historically been patentable as conceiving an idea in one's mind and even expressing that idea in writing does not involve the experimentation on physical matter or any technical effect and can happen instantly. We (except for the EPO with its vested interests) have not thought it fit to extend the patent system to covering intangible ideas, whether expressed on paper or by way of a computer program. As Hartmut Pilch of the Federation for Free Information Infrastructure ("**FFII**") writes:

"*why should it make any difference whether I run [the ideas] in my head, with pencil and paper or with the normal tool of today's civilisation, which is the universal computer…… because the economic rationale behind not granting patents on thoughts applies also here: abstract ideas are, regardless of their applicability to technical problems, produced without experimentation cost, applicable to an infinite range of problems, and replicated at zero cost with no overhead to which patent license fees could be added. The division of the extra cost of patents by the marginal cost (and long-term ideal price) of information goods is a division by zero*."

The patent system was very much designed in relation to physical inventions rather than abstract ideas and has worked well in the industrial and pharmaceutical sectors. A patent grants the inventor a 20 year monopoly right iIn industries which produce technical inventions. This time period has worked well in such industries because often the R&D time needed to arrive at an invention, such as a new drug, can be very significant and also very expensive. In other words, these industries have very long and expensive *development life cycles*. The technology in these industries does not develop rapidly or incrementally. If after 4 years of researching a drug, one knows that you can have a monopoly right for 20 years to exploit the drug this gives drugs companies the incentive to invest in R&D knowing that they can recoup the costs. It is the same for an industrial process. The companies involved in such sectors are often large companies with money to spend on filing patents and defending patent infringement cases which can be incredibly expensive[23].

The role of public policy

As patents for traditional industries may work to generate innovation, an invention which produces physical effects, including a hybrid invention which involves a computer program, should in theory be patentable (save for not foreclosing the software element as discussed above). However, we should not be destroying years of carefully thought out legal, economic and scientific practice which has focused on granting patents for inventions which result in a change to natural forces (i.e. have physical effects) and not computer programs. This is a recognition that inventions which result in a change of natural forces may take many years to develop and may have a beneficial effect on society, hence the need to give inventors an incentive to spend R&D in coming up with such inventions. At the very core of patent law there is a balance between protecting the rights of an individual inventor and those of society. As Peter Prescott QC commented:

"There can be but one justification for having a patent system, and that is that it is good for the people of the country. If the patenting of certain things does more harm than good, it matters. Patents that are wrongly granted can be very expensive to challenge and may deter small and medium enterprises."[24]

Patents grant monopoly rights. Anyone who has studied economics knows that monopolies are not good for competition:

"*A patent system is always a burden on trade, commerce and industry: if only because of the "red tape" effect. The only question is whether the benefits outweigh the burdens.*"[25]

In fact, much of the history of European integration has been based in creating a level playing field in industry by breaking down the negative effects of monopolies so it is quite ironic that the EU Competiveness Council could introduce a directive which would have led to a damaging economic monopoly. Prescott QC is bold enough to admit that patents are not the divine right of inventors but are only granted where they benefit society. It is the citizens of Europe who have the ultimate say in whether we should extend patentability, represented through its democratic institutions and not the powerful voice of minority special interest groups.

There is no doubt that in some fields, society benefits from the patent system because, for fields of technology where the development life cycle is very protracted, expensive and involves a new industrial process, without a system such as patents it would be very difficult to encourage companies to undertake such expensive research and commit the time to developing such new processes (which benefit society) if they did not have the economic security of patents (i.e. a monopoly). In a field such as the drugs industry, where drugs may take many years to develop, a 20 year monopoly on a drug sounds reasonable. However, in software terms, a 20 year monopoly is a lifetime. Furthermore, even if the patent system was amended for software patents (e.g. by shortening the period of the monopoly right), it is submitted that the patent system as a system for protecting and encouraging software innovation is fundamentally flawed.

Do patents work in the software industry?

Traditionally, it has always been understood that the patent system was not appropriate for software. Prescott QC referred to the reasoning at the time of the EPC was drafted and that it was felt *"such patents were not really needed , were too cumbersome and would do more harm than good"*[26]. This "history" is being forgotten but has the position really changed? In *1994*, at the USPTO hearings on software patents a certain software company made the following statement:

"*software per se should not be allowed patent protection…. [We have] built [our] business by creating new markets with new software. We take this position because it is the best policy for maintaining a healthy software industry, where innovation can prosper.*
[...]
when we….founded a company on the concept of software to revolutionize the world of printing, we believed that there was no possibility of patenting our work. That belief did not stop us from creating that software, nor did it deter the savvy venture capitalists who helped us with the early investment. We have done very well despite our having no patents on our original work.
On the other hand, the emergence in recent years of patents on software has hurt [us] and the industry. A "patent litigation tax" is one impediment to our financial health that our industry can ill-afford. Resources that could have been used to further innovation have been diverted to the patent problem. Engineers and scientists such as myself who could have been creating new software instead are working on analyzing patents, applying for patents and preparing defenses. Revenues are being sunk into legal costs instead of into research and development. It is clear to me that the Constitutional mandate to promote progress in the useful arts is not served by the issuance of patents on software".

Who said the above, was it a small time developer or an open source giant? No, this comment was made by one of the largest proprietary software companies of its time, Adobe. This was 1994: perhaps things have changed in the 21st century? At the US Federal Trade Commission's hearings on the anti-competitive effects of patents held in 2002, Bradford L. Friedman, Director of Intellectual Property at Cadence Design Systems, Inc commented:

"*As I'm sure this committee is aware, there is a general animosity to pure software patents within and outside of the industry due to, one, the perceived allowance of what I'll diplomatically call overbroad patent claims, and two, the historically non-proprietary culture of the software engineering industry.*
In sum, largely because the current patent system is poorly fashioned for the software design tool industry, the industry has evolved to minimize the impact that patents have on competition and has relied on other more market-oriented drivers of innovation."

A computer program, i.e., the lines of code, mathematical models etc are all intangible and a computer program, on its own, does not produce any change in natural force (i.e. it does not have a physical effect). Software is an "intellectual creation". As noted above, Article 52(2) of the EPC expressly excludes "computer programs" from being patentable. However, computer programs are protected by copyright, an automatic intellectual property right (there is no need to apply for copyright) which protects the expression (i.e. the code) of a programmer's ideas from being copied by others. Copyright does not, however, seek to protect the ideas (features or functionality) behind a software program. As Prescott QC comments:

"*As was pointed out by Laddie J in Fujitsu Limited's Application [1996] RPC 507, 530, the items listed in Article 52 were excluded for reasons of public policy. This is obvious, for instance, in the case of 'aesthetic creations' – a literary work, say. Such things are to be protected, if at all, under copyright law. Not only does copyright law refuse to protect a general idea, it freely allows the publication of similar works if there has been no copying. Imagine what would happen if literary works could be protected by patent. Literary creativity would tend to be stifled, and authors would have to conduct patent infringement searches before the expiry of their copy-deadlines.*"[27]

You could read "computer programs" where you read "literary works", for the effects are exactly the same. Prescott QC can see the negative affects of patenting literary works, including how this would affect innovation. However, although computer programs are written works which are generally processed by a machine but could be read in source code form, many people fail to see the blinding similarities between the dangers of trying to patent a story plot and the lines of code or functionality of a computer program[28].

The result of the copyright system is that another programmer can come along, *study the ideas* of a computer program and write a better program based on these ideas that does not necessarily infringe copyright. If the software was patented, he could not do this without the risk of facing a patent infringement suit because patents protect the underlying ideas. Copyright protects the hard work and investment made by individuals and companies by protecting the expression of the ideas and has allowed businesses to write extremely profitable software and led to a very strong IT economy in Europe. However, it has also allowed millions of developers to come up with better ways of achieving the same functionality for a specific business process (thus benefiting consumers with more choice and better products) without having to fear being sued for patent infringement because someone owns a patent to a technique, functionality or business process achieved by software. Therefore, copyright has allowed the software industry to flourish, generate numerous new products quickly (without having to worry about patent searches etc) and ultimately benefited consumers. The enforcement of copyright is also much cheaper than enforcing patents, which helps to protect the majority of the software industry, SMEs. The way copyright works suits the way the software industry and software development works. Not every industry is the same and patents are not suited to the software industry, copyright is (or is at least better suited).

The software industry differs markedly from other industries which benefit from patents. The software development life cycle is much shorter than other industries and the costs of R&D and development are much smaller. The way software is also developed is important, software developers have traditionally developed software in an incremental process and the majority of today's software is built on the ideas of previous developers.

"Software clearly differs from other industrial computer products in that it evolves on the basis of pre-existing ideas. This process of open sourcing, which has enabled innovation in the sector to thrive, would become almost impossible under a patenting system." [29]

The Commission on Intellectual Property Rights concluded in its final report in 2002:

"*The patent system fits best a model of progress where the patented product, which can be developed for sale to consumers, is the discrete outcome of a linear research process. The safety razor and the ballpoint pen are examples, and new drugs also share some of these characteristics. By contrast in many industries, and in particular those that are knowledge-based, the process of innovation may be cumulative, and iterative, drawing on a range of prior inventions invented independently, and feeding into further independent research processes by others. ... The development of software is very much a case of building incrementally on what exists already. Indeed, the Open Source Software Movement depends precisely on this characteristic to involve a network of independent programmers in iterative software development on the basis of returning the improved product to the common pool*"[30]

This view was reiterated in a report prepared by Deutsche Bank in 2003:

"[O]ne could be tempted to consider ever stricter IP protection regimes to provide ever more stimuli for innovation.

This conclusion is wrong, however. A prime example is patents on software, which might at first sight be seen as a logical expansion of the classic technology patent. But creating software differs markedly from creating machinery and the like: MIT researchers Bessen and Maskin argue that innovation in software is both strongly sequential (one invention building on a previous one) and complementary (thriving on parallel approaches to the same problem), far more so than in other technology fields. In fact, they found empirical evidence that software patenting substitutes R&D activity, rather than encouraging it, and conclude: For industries like software or computer, there is actually good reason to believe that imitation becomes a spur to innovation, while strong patents become an impediment. In accordance with other academics, they strongly favour copyright over patent protection for software. Copyrighting provides both adequate leeway for sequential innovation and enough protection for marketable software products."[31]

Follow the yellow brick road – only if you pay a toll

Software is written to solve a "problem". You could argue that the "problem" is the functionality, feature or method achieved by the software. The difference with software is that there are many ways in which the *same* "problem" can be solved by writing the software in a different way. Now in a patent system, the "problem" itself can be patentable. This means that once one person has patented the "problem" with a piece of software, no other person can write a piece of software (even though the software may be completely different and not have one line of code the same) without obtaining a patent licence. However, the whole industry revolves around the rapid evolution of software. Being able to write different software to solve the *same* problem leads to better software; if this is prevented or restricted, the result in less innovation, poorer products, less choice for the consumer and less competition (which normally means higher prices for consumers and businesses). There are many paths to the same destination. Some are more hazardous than others. With software patents you gain a monopoly over the destination turning a free motorway into a toll road. What is worse is that not only do you have to pay to get to the destination but you are prevented from reaching the destination by exploring different paths to the toll road even though they may be quicker, more efficient and less costly.

Defeating the purpose of patents

Monopolies are in most economic circles considered as 'bad'. A huge effort has been undertaken within the EU to remove barriers to trade created by monopolies. If a monopoly is to be granted for the implementation of an idea, it should only be done when there is a clear economic benefit to society in doing so. If the granting of patents in a certain field does not encourage innovation then the purpose of patents (as far as society is concerned) is defeated. On software patents, the balance of opinion is that in the software industry software patents have the reverse effect.

The Liberal Democrats in the UK firmly recognise the negative effect of software patents:

"we…..recognize that there are circumstances in which protection systems can have the contrary effect of stifling creativity against the public interest. We are especially concerned about the use of patents in the area of computer software in this respect.

There are usually many ways to achieve the same objective using computer code. The public benefits from the fact that different teams of programmers will work on solving problems and release their separate solutions as competitors in the market. The specific code each team has written is protected by copyright. Allowing a wide definition of inventiveness for patents in the field of software could lead to a reduction in this creative activity. This might be justifiable if there were evidence that the software industry as a whole were suffering because of an inability to secure revenue for research and development but there is no evidence that this is the case as the sector remains vibrant and growing."[32]

However, what better source of evidence then the US whose software industry has had to live with software patents for over 10 years. The internationally respected Federal Trade Commission carried out in October 2003 a detailed industry study into patents. Its conclusions, in relation to software, make fascinating reading:

"*The software and Internet industries generally are characterized by five factors: (1) innovation occurs on a cumulative basis; (2) capital costs are low, particularly relative to the pharmaceutical, biotechnology and hardware industries; (3) the rate of technological change is rapid, and product life cycles are short; (4) alternative means of fostering innovation exist, including copyright protection and open source software; and (5) the industries have experienced a regime change in terms of the availability of patent protection.*

Panelists consistently stated that competition drives innovation in these industries. Innovation is also fostered by some industry participants' use of copyright protection or open source software. Several panelists discounted the value of patent disclosures, because the disclosure of a software product's underlying source code is not required.

Many panelists and participants expressed the view that software and Internet patents are impeding innovation. They stated that such patents are impairing follow-on incentives, increasing entry barriers, creating uncertainty that harms incentives to invest in innovation, and producing patent thickets. Panelists discussed how defensive patenting increases the complexity of patent thickets and forces companies to divert resources from R&D into obtaining patents.

Commentators noted that patent thickets make it more difficult to commercialize new products and raise uncertainty and investment risks. Some panelists also noted that hold-up has become a problem that can result in higher prices being passed along to consumers."[33]

The Government in its conclusions regarding the Patent Office's Consultation Document of 01 November 2000 ("Should Patents be Granted for Computer Software or Ways of Doing Business?") concluded in relation to the relationship between patents and innovation in the software industry:

"*The Government is sympathetic to many of the points made by those concerned about the impact of patents. There is a vast amount of innovative software development taking place without patent protection at present (though, like all software, it is protected by copyright against direct copying). Much of it is being carried out by individuals and SMEs. To extend patentability so that these developers have to divert time and effort into making sure they are not infringing patents, and seeking and enforcing them, would impose a major burden. The necessary case for believing that a significant extension of patentability would increase innovation in this field simply has not been made. In fact, as many respondents suggested, it could have the opposite effect.*"[34]

The rise of the software industry in a patent free world

For 30 years, Europe lived without software patents. What happened in this period? We saw the rise of a great software industry, the birth of the internet and e-commerce, thousands of software products (and the rise of **F**ree and **O**pen **S**ource **S**oftware ("**FOSS**")). The Internet has enriched all our lives but had their been software patents in place at the time of its inception it is very unlikely that we would have achieved so much for many forget that the Internet has been built on open standards:

"*Of course the main internet interfaces are open, based around freely available published standards that are not under the control of any one company or industry body.*"[35]

We should all take note that the father of the world wide web, Tim Berners-Lee, is a vocal opponent of software patents. Do we really think that the idea of an electronic shopping cart should be monopolised to the tune of US$40,000,000? This is what Amazon has had to pay to secure the continued use of its e-commerce platform in the US this year[36]. It would not need to pay this in Europe. Imagine every other e-commerce vendor in Europe who uses an electronic shopping cart. Would they have the funds of Amazon to be able to negotiate a patent settlement with the company that sued Amazon? Perhaps this is not an example close to lawyers' hearts. What about a rather fruity story turned sour with a disputed settlement of US$450,000,000? This is the cost (although the saga is still running) that the makers of the Blackberry handheld e-mail device, Research in Motion ("**RIM**"), offered to NTP, an American software company, to make NTP stop its patent infringement claims based on very wide software patent claims successfully obtained by NTP in the US. The latest, on this long running battle, is that RIM could end up being forced to shut down its US operations[37] – that's right, NO BLACKBERY. Do not fear though, thanks to the desire of some within Europe to preserve innovation rather than profiteering from the monopolisation of intellectual processes, you can still continue to get your daily dose of thumb arthritis and access to your office e-mails. This wonderful business tool is alive and well in Europe because, you guessed it, we have resisted a move towards a US style system of software patents. Even if the little berry wins through, the sheer cost of fighting such a battle will have hurt RIM's balance sheet, its investors and, if it is forced to pay an even higher settlement, the battle may ultimately hurt the wallet of business users. These are just a couple of examples from the software patent horror show illustrating how software patents affect companies' bottom line, hurt innovation and ultimately affect us, the consumers.

Did the absence of patents prevent innovation? Absolutely not. Software companies knew that they could, as developers had been taught, study the ideas and industry programming techniques and, provided they did not steal someone else's expression of those ideas, write a program without the fear of being sued for patent infringement. Developers knew that as long as they created an "independent" work, they would potentially stand to profit from it. There was no need to do exhaustive searches of patent applications, nor a danger of being liable for patent infringement for using a technique which you had independently created. It is in this environment that FOSS has flourished.

FOSS: a benefit for all

FOSS often relies on thousands of developers from all over the world contributing to a software project. FOSS projects, as a result, have very fast development times. Not only this, the products are normally very stable because they have such a huge testing base. FOSS, which is often free to purchase or where the costs of purchase are very much reduced, is being used in schools[38], hospitals[39], local authorities, small businesses and large corporations. It is giving technology to the masses and is vital for the development of the third world. For many small businesses, it allows them to keep their costs down and focus on delivering products to their customers at reduced prices because FOSS lowers the cost entry barriers. Ultimately, FOSS is helping not-for-profit bodies, government bodies and most importantly, everyday citizens to harness the full benefits of technology and software at little or no cost.

Damaging FOSS would be to damage a market model which lowers costs whilst increasing productivity. The "open source" business methodology has huge potential not just in software but in other business fields[40], allowing for innovation to take centre stage through collaborative research projects and harnessing the input of the many in an open rather than closed environment. A hammer blow for FOSS could terminate this embryonic alternative business model.

If we allow in Europe, through careless drafting or otherwise, software patents, this may destroy the FOSS movement and with it much software innovation. This is not an exaggeration. If each and every developer from around the world contributing to a FOSS project is not free to use the ideas behind software programs, or a technique or feature because it is patented, the fundamental building blocks of software development will have been taken away from them. Furthermore, the fear of patent infringement may mean that many developers will not be able to contribute to FOSS projects because of the huge expense of defending a possible patent infringement claim. Even if they tried (which would take time and therefore take away from development time) to carry out a search of patent applications they would not necessarily understand if they were infringing a patent because patent claims are written "in a foreign language", full of legalese and it can be unclear as to what is covered or not by the patent. These small developers, who have made a massive contribution to FOSS software, have not got the time or money to pay expensive lawyers to determine whether they fall within or outside of a particular patent.

A thorn in the side of innovation

What is known by millions of businesses in Europe is that software patents are bad for the software industry *as a whole*. Even the UK Patent Office, which carried out a consultation on software patents and business method patents received an overwhelming response, which its conclusions noted, that software patents did not necessarily result in more innovation but could have the reverse effect. If software patents are allowed, US and Japanese companies will immediately precede to registering patents in Europe to match their US patents (many of which have a dubious legal basis) and which represent the obvious building blocks and techniques of software developers which in a patent free world have been capable of being used by everyone free of charge for the greater good of the software community and ultimately consumers. Once registered, no doubt a string of law suits or requests for royalties will be made against IT SMEs which form the backbone of the software development industry in Europe, resulting in many being made bankrupt or having their profit margins slashed with the consequence of lower investment and development for the future (hence less innovation).

A knock on consequence will be SMEs having to divert away resources from development to having to investigate patents before development work can be undertaken and also having to defend patent infringement claims. This will mean less development time, greater time to market of products, greater TOC (Total Cost of Ownership) and ultimately less innovation. This is not a claim without substance for it is exactly what has happened in the US. In 2002, at the FTC hearings on the effects of patents on innovation, the CEO of Divx Networks, R Jordan Greenhall talked about the frustrations of one of his lead developers trying to work out whether one of their technologies infringed another patent:

"*The poor guy spent the better part of five days examining all these different patents and came back to me saying, "I haven't the slightest idea whether or not we infringe on these patents, and frankly, they all seem to infringe on one another." The end result being that I have no idea whether my product infringes on upwards of 120 different patents, all of which are held by large companies who could sue me without thinking about it. The end result, much like Borland, I have now issued a directive that we reallocate roughly 20 to 35 percent of our developer's resources and sign on two separate law firms to increase our patent portfolio to be able to engage in the patent spew conflict. I think the concept here would be called saber rattling. I need to be able to say, "Yeah, I've got that patented too, so go away and leave me alone.*"

The software industry has thrived without patents, with numerous products developed quickly because developers do not have to worry about obtaining patent clearance for the numerous techniques they use in writing a program. This did not go unnoticed for Prescott QC who commented:

"*it is worth noting that the software industry in America developed at an astonishing pace when no patent protection was available*."[41]

Under a software patent system, these techniques will no doubt be patented in Europe because of the EPO approach to allowing software patents and will result in the US style system where the very building blocks of normal software development are ring fenced by the big corporations, which have the resources enabling them to spend time and money to file hundreds of patent claims and in the process laying stake to the very core of a developer's usual toolkit and preventing developers (without having to pay hefty licensing fees) from developing innovative and better products. Even Japan, a historic supporter of software patents, has started to realise that they may not be the right way forward and could deter future innovation.[42]

A defensive battle - the patent shield

However, once we accept a patent system for software, it will become very difficult to get out, for companies will have no choice but to try and get patents and built up a patent shield – this is referred to as defensive patent filing and is prolific in the US:

"*There is absolutely no evidence, whatsoever -- not a single iota -- that software patents have promoted or will promote progress.*
[...]
The company for which I am speaking, Autodesk, holds some number of software patents and has applied for others -- which, of course, remain secret under current U.S. law. However, all are defensive -- an infuriating waste of our technical talent and financial resources, made necessary only by the lawyer's invention of software patents. Autodesk has faced at least 17 baseless patent claims made against it and has spent over a million dollars defending itself, with millions more certain to pour down the bottomless patent pit unless we halt this debacle. Fortunately -- unlike smaller software producers -- we have the financial and technical resources to rebuff such claims. We rebutted all but one of the claims, even before the patent-holders could file frivolous law-suits, and will litigate the remaining claim to conclusion. Note that [the USPTO] has issued at least 16 patents that we have successfully rebutted, and we never paid a penny in these attempted extortions that [the USPTO] assisted.
But it was an enormous waste of resources that could have better been invested in useful innovation. These unending baseless claims benefit patent lawyers, but they certainly do not promote progress."[43]

So after you have built up your own "patent thicket", you have a shield with which to defend oneself from patent attacks. Therefore, several of the American IT giants that got on their knees to the USPTO in 1994 to ask them not to allow software patents have, after over 10 years of building up their shield (at considerable cost and at the expense of development budgets, hence innovation) less of a resistance to them. However, this cannot be said for every single new entrant or for all the existing European SMEs who do not have such a shield (and who represent the majority of the IT industry in Europe).

Will Europe benefit from software patents?

Who will be the main beneficiaries of a EU patent system where software patents are allowed (whether expressly or through bad drafting and/or EPO policy)? Will it be the millions of small IT companies in Europe, the European economy or the citizens of Europe? No, the beneficiaries of having software patents or an unclear community patent regime will be the US and Japanese economies and the few large IT corporations (most of which are not European owned). Having software patents will go against the very intention of the statements made by Viviane Reding, European Commissioner for Information Society and Media, to increase research and innovation in the European ICT sector and the statements from the policy consultation document for 2010i strategy

"*For content in general, the EU should use a copyright system instead of a patent system but not one that will restrict user rights*"[44]

For the big IT companies, software patent thickets are built up and cancelled out through cross-licensing arrangements. So what is the actual benefit, as clearly innovation is not the winner? Well, it is the fact that the big companies with large software patent portfolios can keep out of the market new entrants and small companies by threatening or actually suing them for patent infringement. In other words, actually destroying future innovation and market choice. As you may recall HP, Microsoft and Apple all had very humble origins. A software patent system is potentially threatening the birth of another giant which could bring innovation and better products to Europe's citizens.

If we are clumsy and let those who do not have Europe's best interests push through the EU institutions a community patent regime which ushers in software patents in Europe (on a legal footing), we will no doubt be driving a stake straight through the very heart of information technology innovation in Europe.

A return to a public policy centric view – wishful thinking

There does finally appear to be a long overdue backlash, if not limited, to the relentless march of intellectual property rights to the detriment of society and the global economy. Apart from the vocal opposition to the CII Directive, in the UK, the historic Royal Society for the encouragement of Arts, Manufacturers & Commerce has recently published the "Adelphi Charter on Creativity, Innovation and Intellectual Property" (the "**Charter**"). The Charter calls on governments to, inter alia, adopt a public interest test in considering new laws on intellectual property requiring "*a balance to be struck between the monopoly rights implicit in intellectual property laws and the free competition that is essential for economic and creative vitality*."[45] As noted above, the Japanese government is also starting to question the effect of software patents on innovation.

The role of SMEs

Myth 5 – SMEs support software patents

The upholding of this myth in the run up to the CII Directive was critical to those either openly, or through the promotion of the Common Position, championing software patents. Why? Well, there are two reasons, before and during the debate, it has been almost universally documented that software patents have a negative effect on SMEs (as noted above). Secondly, and this is less known, it is not Microsoft or IBM and the like that are the key to the European IT industry, it is the thousands of IT SMEs which make up the majority of the European IT industry. More importantly, and central to the whole patent debate, it is the SMEs which are the key drivers of innovation (and hence prosperity and development) in Europe. However, this is not just my conclusion, this is the conclusion of those responsible within the European Union for driving innovation in the high-tech sector:

"*Small and medium-sized enterprises (SMEs) play a crucial role in European competitiveness and job creation, not only because they represent the overwhelming majority of enterprises in Europe, but also because they are the source of dynamism and change in new markets, particularly those at the leading edge of technology*".[46]
.
If innovation is truly the reason why the pro-software patent lobby want software patents (which I doubt), they have a fundamental hurdle to overcome. The key group within the IT sector, SMEs has been shown time and time again to suffer most from the introduction of software patents and since they are the driver of innovation, a negative impact on SMEs results in the stifling of innovation. Hans-Werner Müller, the Secretary General of the UEAPME, an organisation which represents 11 million small businesses, had the following stark view of what the CII Directive in the form proposed by the Commission et all meant for European SMEs:

"*This directive will threaten the existence of many small businesses if passed in its present format".*

40,000 and counting…

Myth 6 – Europe has over 40,000 software patents and the IT industry continues to thrive

This is a classic and particularly misleading argument used by the pro-software patent lobby. They argue that the EPO has granted 40,000 plus "software invention" patents and point to how the European IT industry has grown in this time, the upshot of their argument being that patents don't affect innovation. This is an incredibly misleading argument. The negative effect that software patents have on the software industry has not yet manifested itself in Europe precisely because of the dubious legality of the software patents granted by an administrative body, the EPO. For software patents to work their destructive magic, they must be no doubt as to their legal status yet in Europe not one software patent has been successfully enforced. Therefore, without "legal bite" in Europe (compare this to the US situation of the very real threat of triple damages for infringement) these software patents have been worthless pieces of paper - they would have become very valuable had the Common Position been accepted since the EPO's unlawful extension of the patent system would have received the formal seal of approval - which could not be used by IT companies to demand licence fees and/or block new entrants to the market. On the other hand, where such pieces of paper have legal value, e.g., the US, the concerns of those against software patents have all materialised and continue to get worse and worse every day the current system is not amended.

Tripping us up

Myth 7 – TRIPS and article 27

An argument used frequently to convince us that we have no choice to accept software patents is that TRIPS requires us to allow patents for all fields of technology. However, this argument is fundamentally flawed for it requires us to accept that "technology" includes software. As noted above, software is not "technology" for patent purposes and is expressly excluded from the scope of patentable inventions in Article 52 EPC. The weakness of this argument was recognised by the Comptroller General of the UKPO as far back as 1997:

"*Some have argued that the TRIPS agreement requires us to grant patents for software because it says "patents shall be available for any inventions.....in all field of technology, provided they are.....capable of industrial application". However, it depends on how you interpret these words.*

Is a piece of pure software an invention? European law says it isn't. Is pure software technology? Many would say no. Is it capable of "industrial" application? Again, for much software many would say no.

TRIPS is an argument for wider protection for software. But the decision to do so should be based on sound economic reasons. Would it be in the interests of European industry, and European consumers, to take this step?"[47]

From the evidence before us, be it economic or, more importantly the immoral nature of monopolising abstract intellectual processes, the verdict is clear, it would not be a benefit to extend "technology" for patent purposes to cover software, it would not benefit European business or its citizens.

Europe as a leader or as a servant?

If we investigate many of the pro-software patent associations claiming to be acting for the software industry, many are either directly or indirectly funded by a small number of very powerful (and non-European) IT companies whose track record on open market politics is not impressive. These companies are extremely large organisations, which have struggled to innovate against much leaner SMEs and, their religious zealotry for software patents is ultimately a recognition that they cannot survive in an open market but instead need to close the market down through the monopolisation of the ideas behind software engineering. Allowing software patents in Europe would open the door to the continued land grab of software ideas by such non-European corporates, legitimise the 40,000 plus software patents already granted by the EPO for the benefit of the non-European IT giants at the expense of the majority of the European IT industry.

The bottom line for IT lawyers

Software patents will only advantage a small group of the largest (and in the minority) IT companies. As IT lawyers, how many of us can say that we have such clients on our books? The bread and butter for IT lawyers are IT SMEs – for such clients, patents are not the solution but the problem. The legitimisation of software patents across Europe would make patent agents better off, as well as certain IP lawyers. However, the long-term picture is very different. Software patents increase the barriers for new entrants, your future client base, and potentially the next Microsoft or Apple (neither would have been able to grow so impressively if a patent rich environment had existed in the early 80s). Furthermore, they put at risk the very well-being and profitability of your existing clients. Patent litigation is extremely expensive. Dealing with patents takes away time from creating innovative applications that generate revenue . With fewer innovative products, the full creative potential of a client may not be realised and if its growth is hindered through lack of innovation or if they are ruined through patent litigation, you are likely to miss out on the long-term benefit of a consistent and, potentially increasing, revenue stream. What is worse is that the effect of patents is to lead to a grouping of ideas with an ever decreasing pool of companies resulting in a move towards monopolisation and fewer but bigger IT companies (which means less clients to go round). Finally, as a consumer of software (which all law firms are), IT costs are likely to increase through less choice, less innovation and less competition and potentially higher prices. Food for thought?

"Let us stand on each others' shoulders, rather than on each others' toes"[48]

Ultimately we have to ask ourselves what kind of software industry we want in Europe? Do we want an industry dominated by a group of companies, maybe even a couple of hundred companies, who have the resources or desire to carve up the building blocks of software engineering where control and monopolisation through patent revenues becomes more important than innovation, or do we want the software industry we have now in Europe, where developers are free (without copying code) to work with these building blocks to create innovative software products for the benefit of Europe as a whole.

With the Community Patent, we have an opportunity to get involved in determining which path we choose, the toll road or the freeway. There remains a need to clarify the position on software patents in a way which does not damage the European IT industry. A good starting point, which has already considerable cross party support amongst MEPs, is the 21 amendments proposed to the Common Position by Mr Rocard (ex-prime minister of France), Mr Buzek (ex-prime minister of Poland) and Andrew Duff (UK Liberal Democrat MEP)[49]. The Common Position as amended by such amendments would have helped rein in the EPO and protected our industry from unnecessary damage.

Copyright or wrong?

The real challenge for IT lawyers and the rest of the industry should be looking at ways we can address the problems (yes there are some problems) with the copyright system rather than putting our weight behind a patent system which hinders rather than fosters growth in the industry we work in. Although, copyright reform is a subject for a whole article in itself, one possible reform would be to create a European registry of software to allow copyright owners to deposit their works at a European level. This would not be intended to change the substantive laws on how copyright arises but would aid certainty and serve as formal evidence of the existence of such rights. Such a register would not be compulsory but optional. This type of registry exists at a national level in Italy and functions on exactly the basis outlined above.

I leave the final word to someone who is in a better position than me to understand the damaging effects of software patents on the software industry, Ron McQuaker, a recognised figure in the British computer industry and past president of the British Computer Society who commenting on the negative effects of software patents concluded:

"*These considerations lead me to the view that a properly applied copyright regime is much more appropriate to the protection of Intellectual Property in programs than patent. You can only breach copyright by doing something morally wrong, i.e. copying, and copyists know whether they have done so or not. The case for grants of monopoly by patent looks to some people rather like an example of pulling up the ladder behind them by the big battalions.*"[50]

The law is stated to the best of my knowledge as at 9 November 2005

Cristian Miceli is an IT lawyer and founder of Lawyers Against Software Patents (LASP) – www.lasporg.info. The article is written in a personal capacity. He can be contacted at: lasporg@googlemail.com

[1] The High Court cases where: (1) IN THE MATTER OF Patent Applications GB 0226884.3 and 0419317.3 by CFPH L.L.C. [2005] EWHC 1589 (Pat); (2) Halliburton Energy Services Inc v Smith International (North Sea) Ltd and others [2005] EWHC 1623 Pat; (3) IN THE MATTER OF Patent Application GB 0017772.5 by Shopalotto.com Limited, [2005] EWHC 2416 (Pat); and (4) IN THE MATTER OF UK Patent Application no.GB0108683.4 in the name of Cecil Lloyd Crawford (2005), 4 November 2005. For the hearing officer's decision, see the decision O/255/05, in the application by Oracle Corporation on 14 September 2005. The UKPO patent notice can be found at <http://www.patent.gov.uk/patent/notices/practice/examforpat.htm>

[2] Per Stephen Kinsella, writer, US patent attorney and IP lawyer. For further details see <http://www.lewrockwell.com/orig/kinsella7.html>

[3] Per James Boyle, Professor of Law, Duke University cited in The Economist, "Free Ideas", 13 October 2005.

[4] LobbyControl and Corporate Europe Observatory wrote to Commissioner Siim Kallas on 28 June 2005. See <http://www.corporateeurope.org/lobbycracy/prc4c.html> for further details.

[5] Paragraph 56 CFPH, emphasis added

[6] As Prescott QC comments in paragraph 14 CFPH with reference to the word "technical", "*it should be remembered that it was not used by the framers of the Patents Act 1977 or the European Patent Convention when they wanted to tell us what is or is not an 'invention*'".

[7] For a good list of such software patents see <http://swpat.ffii.de/patents/samples/index.en.html>

[8] Paragraph 45 CFPH, emphasis added

[9] Paragraph 97 CFPH

[10] Paragraph 9 Shopalotto.

[11] BPatG Error Search 2002/03/26. Emphasis added

[12] BGH 1976-06-22: Dispositionsprogramm. Emphasis added.

[13] Paragraph 25 CFPH, emphasis added.

[14] As Prescott QC commented at paragraph 27 CFPH, "*the Convention is expressed in three languages, all equally authentic. It is therefore not surprising that the methods of interpretation to be applied to such an international instrument are not the same. In particular, there is more room for a teleological interpretation*."

[15] Paragraph 35 CFPH, emphasis added.

[16] Paragraph 104 CFPH, emphasis added.

[17] Ibid. footnote 1

[18] Paragraph 104 CFPH.

[19] Paragraph 37 CFPH.

[20] Although Prescott QC uses 'as such', he makes it clear earlier in his judgement that he doesn't approve of excluded subject matter being patentable by claiming some form of "physical" artefact and that "technical" has to be construed with reference to Article 52 EPC

[21] Report to the European Commission by Robert Hart (Independent Consultant), Peter Holmes (School of European Studies, University of Sussex) and John Reid (IP Institute) on behalf of Intellectual Property Institute, London 24 July 2001. Emphasis added.

[22] Per Hartmut Pilch, FFII.

[23] Amazon's profits were badly hit this year paying US$ 40,000,000 in relation to a patent claim (relating to software!). See <http://news.com.com/Amazon+pays+40+million+to+settle+patent+dispute/2100-1030_3-5829193.html>

[24] Paragraph 11 CFPH, emphasis added.

[25] Per Prescott QC at paragraph 35, CFPH.

[26] Paragraph 35 CFPH, emphasis added.

[27] Paragraph 30 CFPH

[28] For a good article comparing the patenting of software to the plot of books, see Richard Stallman's article on Guardian Online entitled, "Patent Absurdity",20 June 2005, <http://technology.guardian.co.uk/online/comment/story/0,12449,1510566,00.html>

[29] UEAPME 27 April 2005 press release. The UEAPME is an industry body representing over 11 million SMEs. Emphasis added.

[30] See <http://www.iprcommission.org/papers/text/final_report/reporthtmfinal.htm>

[31] See <http://www.dbresearch.com/PROD/DBR_INTERNET_EN-PROD/PROD0000000000175949.pdf>

[32] See the Liberal Democrats IT policy document at <http://www.makeitpolicy.org.uk/PP-ISociety-Copyright.html>. Emphasis added

[33] "To Promote Innovation: The Proper Balance of Competition and Patent Law and Policy", A Report by the Federal Trade Commission October 2003, Conclusion (Chapter 3, V.G, pages 164-165). See <http://www.ftc.gov/os/2003/10/innovationrpt.pdf>. Emphasis added.

[34] See paragraph 15, "Should Patents be Granted for Computer Software or Ways of Doing Business? The Government's Conclusions", <http://www.patent.gov.uk/about/consultations/conclusions.htm>. Emphasis added

[35] Per, Bill Thompson, journalist and technology commentator and frequent contributor to the BBC technology news

[36] Ibid footnote 23

[37] For a good article on the latest position see, <http://searchmobilecomputing.techtarget.com/originalContent/0,289142,sid40_gci1149116,00.html>. What is worse in the Blackberry story is that the USPTO may potentially end up ruling that NTP's patents are invalid (not because they are software patents but because of prior art) yet because of the court process RIM may end up having to settle anyway to avoid an injunction notwithstanding the patents may be revoked in due course. On 5 December 2005, a further NTP patent was revoked. For more details see, <http://www.theregister.co.uk/2005/12/05/uspto_rejects_ntp_patent/>

[38] BECTA, (the British Educational Communications and Technology Agency), conducted a study which was published earlier in this year in which confirmed the important role that FOSS could play in schools in Britain. See <http://www.becta.org.uk/corporate/press_out.cfm?id=4681>. Also see, Tom Espiner's article, "Italian schools move to Linux". ZDNet UK, September 05, 2005, <http://news.zdnet.co.uk/software/linuxunix/0,39020390,39216134,00.htm>

[39] Novell has just won a contract to supply the NHS with Novell's open source based SUSE Linux platform. The NHS expects to save approximately £75,000,000 over three years. See <http://www.theregister.co.uk/2005/12/06/novell_nhs_win/>

[40] For examples of interesting alternative applications of "open source" business methodology see the "Danish" beer example, <http://news.bbc.co.uk/2/hi/technology/4718719.stm>and the Open Source Biotechnology project, <http://www.bios.net/daisy/bios/15>

[41] Paragraph 35 CFPH.

[42] Interim Report of a study group set up by the Japanese Ministry of Economy, Trade and Industry delivered in June 2005 as cited on "Japan Today " – see <http://www.japantoday.com/e/?content=news&cat=4&id=351826>

[43] Per Jim Warren, board member of Autodesk at the USPTO 1994 hearings on software patents. Emphasis added.

[44] Final Report, Public Consultation on the new Information Society Strategy beyond 2005, 26 November 2004 – 17 January 2005. Pg. 11 and 12 <http://europa.eu.int/information_society/eeurope/i2010/docs/2010_challenges/050210_consultation_final_report.doc> Emphasis added.

[45] Per the RSA's press release dated 14 October 2005, "International Commission calls for Governments to adopt new public interest benchmarks for intellectual property".

[46] Report entitled "Support To The Participation Of SMEs In The Sixth Framework Programme", 18 Dec 2002, ftp://ftp.cordis.lu/pub/documents_r5/natdir0000036/s_1891005_20030127_130002_6L021891en.pdf . Emphasis added

[47] Per Paul Hartnack, in his speech as chair of the "Software Patents in Europe" UKPO conference held on 23/03/1998. Mr Hartnack was the Comptroller General of the UKPO at the time.

[48] Per Jim Warren, board member of Autodesk at the USPTO 1994 hearings on software patents

[49] See <http://swpat.ffii.org/papers/europarl0309/amends05/komprom0506.en.pdf>

[50] Ron McQuaker, past president of the British Computer Society in his speech at the conference cited at footnote 45 above.