BEYOND SOFTWARE PATENTS

JOSE R. TRIGUEROS

ABSTRACT. Software patents are still a heatedly international debate. Intellectual Property scholars of almost every country discuss if software should be protected by patents or copyrights while software evidently have elements of both “creation” and “expression”. Such a discussion should be surpassed already and instead such energies and ink should be focused on legal and administrative issues such as patent examination and standardization among others. This study is focused on giving alternatives to countries which still issue patents for computer implemented inventions in order to expedite the acceptance of software patents.

1. Introduction

Software has hastily become an essential component of our daily life1. From the handy blackberry to the refrigerator we deal with devices that are programmed to perform and achieve goals that suit our needs. Much like the first airplane and the early development of modern aeronautics, the code became embedded to our society and is still being analyzed in different aspects of our culture2.

One of them is the legal aspect, which has struggled to determine where could software fit to be duly protected and even if it should be protected at all3. From the standpoint of legislation, courts and precedents, the United States and Japan have developed and taken the lead in the field of software protection, allowing the access to patents for the software development sector.

---

1 The demand for computing services has been steadily growing in almost every field of industry. Such a trend has also reached government institutions, schools, universities and home. Please see R. P. MERGES, P. S. MENELL & M. A. LEMLEY, “Intellectual Property in the New Technological Age”, Wolters Kluwer Law & Business, U.S. 2007, pp. 972.

2 The amount of scholarly research in the field of software patents is steadily growing and it will surely continue to expand, attending to the wide array of topics that need to be addressed not only at a national but also at an international level.

particularly in the United States where the decision was at the least, controversial\textsuperscript{4}, it nonetheless shred some legal certainty at the time as patents became fully accessible for software development companies, which in consequence led the European Union\textsuperscript{5} as well as other countries to try to determine the economic and social advantages of such decision\textsuperscript{6}. However, several players in the sector did not agree with the resolution. Following such trend several scholars and a strong community of software developers around the globe fiercely argued against what they perceived as a direct attack for what should be considered a public good\textsuperscript{7}.

Nonetheless, is it still necessary to continue discussing which kind of protection should be more appropriate for software, if any, when more pressing issues are at hand now? This analysis has as starting point the assumption that software is an invention which should be protected by means of patents, while certain aspects of it are still being covered by copyright\textsuperscript{8}.

The study is divided in two chapters: The first one provides a brief outline of the development of patent protection for software in the United States, resuming how the different U.S. Courts have framed the legal patent system in more than 30 years of experience dealing with this particular kind of inventions. As the study is

\begin{itemize}
\item[4.] The development of seminal cases which lead the US courts to the acceptance of software patentability reached its peak with State St. Bank & Trust Co. v. Signature Fin. Group, Inc., 149 F.3d 1368, 47 U.S.P.Q.2d (BNA) 1596 (Fed. Cir. 1998), cert. denied, 119 S. Ct. 851 (1999).
\item[6.] It is worth mentioning that conclusive results regarding the benefits of patent protection for computer programs have not been reached yet. Please see K. BLIND, J. EDLER & M. FRIEDEWALD, “Software Patents: Economic Impacts and Policy Implications”, Edward Elgar Publishing, U.K., 2006, see also R. J. MANN, “Do Patents Facilitate Financing in the Software Industry?”, supra note 3.
\item[7.] Along the discussion topics as relevant as freedom of access of information and the shrinking of free commons have also been thoroughly discussed. Please see E. MOGLEN, “Anarchism Triumphant: Free Software and the Death of Copyright”, in First Monday, August 1999, available at (http://emoglen.law.columbia.edu/publications/anarchism.html) last visited on October 03, 2007, see also L. LESSIG, “The Architecture of Innovation”, the lecture was delivered at the Inaugural Meredith and Kip Frey Lecture in Intellectual Property at Duke University School of Law on March 23, 2001.
\item[8.] Please see R. WATT, “Patent and/or Copyright for Software: What has been done so far?”, in Review of Economic Research on Copyright Issues, 2007, pp. 3-14.
\end{itemize}
mainly based on the U.S. development on the issue, the purpose is to later point out the advantages and mistakes that such experience provides.

Chapter two covers the advantages of using the U.S. experience as a starting point for other countries with patents for computer implemented inventions as well as strategies for avoiding U.S. mistakes, in order to -paraphrasing Mr. Thomas Alva Edison- start where the last country left off.

Important issues to study are the difficulty of prior art searches, examination regarding novelty, non obviousness and industrial application and their extent regarding software patents, the issue of submarine and petty patents linked with litigation treats and costs.

This chapter will also deal with proposals in the patent law framework in order to avoid prosecution inconveniences, dealing also with the issue of standardization in the software industry.

As mentioned above, the purpose of the article is to leave behind the discussion regarding software patents in countries aside Japan or the United States and focus such energy in addressing both legal and administrative issues that are now affecting software development firms and the economic development of a very important sector vital to other fields of industry.

2. What are we dealing with?

Software surrounds us in an insurmountable amount of different devices that come with us while traveling, in the office and in our home. Yet the definition of software seems elusive not only from a legal standing but also for the day to day user. The term “software” is often associated with several different meanings⁹ and thus, in the absence of a clear definition for such technology the protection and interpretation may give rise to unexpected results from attorneys, judges and

---

policymakers, which could reach different conclusions when the issue of software protection becomes present\textsuperscript{10}.

It is also elusive the way software should be protected. However, the development of patent protection for software in the U.S. Courts has not only reached a satisfactory conclusion regarding an adequate protection in accordance with the patent law, but also regarding its definition\textsuperscript{11}.

Computer programmers begin developing what would later be called as software as we know it during the early 50’s\textsuperscript{12}, and although the methods for copying software were not massively available as they are nowadays, the protection of computer programs was a need among the sector. However, at the dawn of what would soon become the digital age, the law was not prepared to offer the required protection for such a new technology. The inherent effect that software caused among legal systems was adaptation and further development.

In the meantime, researchers and early software developers freely shared code and information, becoming an active source for future breakthrough inventions. Information flowed easily and without boundaries within a blossoming development community that was not interested or restricted by the rule of law.

However, at that early stage computers where not mass market accessible, more limited to companies and researching institutes. Software was profitable but not as it would soon become with the advent of the digital age, when lots of homes would turn to buy computers and digital encyclopedias\textsuperscript{13}.

\textsuperscript{10} In light of the number of definitions available in the scholarly studies mentioned in the preceding footnote, it should be stressed that some of them could be considered overly broad, encompassing software and also additional goods while others could narrow the scope of what software is, excluding rightfully defined software applications.


\textsuperscript{13} Please see G.A. STOBBS, “Software Patents”, supra note 9, at 133.
An enormous market arose to software development companies when people begun requesting more accessible and less technical applications to work, teach and have fun on a personal computer. Software became embedded to our society, from small children playing with interactive teaching software to adults playing virtual poker.

The economical relevance of software became a bright spot on those software developing companies. Easy access and free share could cut profits and produce undesired competitors and thus, an invention that was almost non proprietary at the beginning became a highly valued proprietary asset, producing thousands of companies, jobs and new development\textsuperscript{14}.

Initially, software owners and their specialized legal counsel opted to protect computer programs by means of contracts and trade secrets\textsuperscript{15}. For several years the protection of software rested in contractual relations, confidentiality agreements and trade secret law mostly due to the fact that software was generally available only for companies\textsuperscript{16}, the development in the digital field was not as advanced as it would be two decades later and thus, illegal copying was not as big an issue as it is now with the advent of the personal computer and advanced digital copying systems.

Trade secret law protection was substantially enough notwithstanding that reverse engineering of computer programs would have been enough to vanquish such protection since software was not mass market designed, rendering contracts as a good tool in order to avoid disclosure or non-authorized copying\textsuperscript{17}.

However, price reductions and development soon began to make personal computers available to the public, software began a slow transition from tailor designing for companies to a more “prêt a porter” mass marketing\textsuperscript{18}. The market of

\textsuperscript{14} The U.S. Software Industry shows a great amount of growth and has steadily opened more job positions. Please see R. J. MANN, "Do Patents Facilitate Financing in the Software Industry?", supra note 3.


\textsuperscript{16} Please see R. P. MERGES, P. S. MENELL & M. A. LEMLEY, "Intellectual Property in the New Technological Age", supra note 1 at 980.

\textsuperscript{17} \textit{id} at 980

\textsuperscript{18} \textit{id} at 973
software and personal computers soon turned to be highly lucrative; the digital revolution started and is, needless to say, still running at an outstanding speed.

Evidently, the old contracts/trade secret law frame of protection would be left aside with the push of the upcoming digital revolution\textsuperscript{19}, and new forms of protection for computer software were considered.

Since software was presented in the first instance as source code and/or object code to attorneys and judges in the early stages of computer development, it would seem fairly reasonable to determine that a set of written instructions would fulfill all the characteristics of a literary work and thus, copyright should be awarded\textsuperscript{20}. Moreover, a new technology which is developed in simple text may not seem as a technical development in the absence of steel and gears, in comparison with mechanical or even chemical inventions of the 60’s or 70’s\textsuperscript{21}.

An additional incentive for such protection was the absence of examination regarding novelty, non-obviousness and technical applicability as requisites for protection, and the cherry on the cake was that copyright was widely accepted throughout the world and enjoyed a fair amount of harmonization. The stage was set for the copyright protection of software\textsuperscript{22}.

Nonetheless, software is a product of computer science and thus its creation, nature, development and behavior may seem extremely elusive for people involved in using software through a physical platform such as a personal computer. People do not just wake up one morning and decide to write thousands of pages

\textit{\footnotesize{\textsuperscript{19} id at 973}}


\textit{\footnotesize{\textsuperscript{22} As stated by the U.S. Supreme Court in BAKER v. SELDEN, 101 U.S. 99 (1879), “The difference between the two things, letters-patent and copyright, may be illustrated by reference to the subjects just enumerated. Take the case of medicines. Certain mixtures are found to be of great value in the healing art. If the discoverer writes and publishes a book on the subject (as regular physicians generally do), he gains no exclusive right to the manufacture and sale of the medicine; he gives that to the public. If he desires to acquire such exclusive right, he must obtain a patent for the mixture as a new art, manufacture, or composition of matter. He may copyright his book, if he pleases; but that only secures to him the exclusive right of printing and publishing his book. So of all other inventions or discoveries.”}}
of logically structured instructions for a computer in order to produce a useful result, for as simple as it may appear to people skilled in the area of computer programming\textsuperscript{23}.

Copyright historically has been the area of Intellectual Property where authors in the field of arts such as literature, painting and sculpture may enjoy protection of their respective works. A work of art requires protection against slavish copying and reproduction just to mention common copyright issues, in order to allow its author to enjoy the profits arising from licensing or assigning the economic rights when desired, at the same time sending the message to other authors of the existence of safe harbor for their work, incentivizing cultural development and concurrently granting the society reasonable means of access to such development.

Consequently, copyright may not be suitable for the protection of technology which, as software, may be easily copied or invented around\textsuperscript{24}. The issue of not suitable protection of software under the copyright regime has been analyzed in several articles in the past and the U.S. has solved such imperfection by granting a double layer of protection\textsuperscript{25} which is stronger and more suitable for protecting the development of an industry which is now producing revenues in the billions of dollars annually.

Yet again, a further enhancement of the protection offered to software developers was needed.

\textbf{2.1 Chronicle of an announced patentability}

During the early days of software development and protection, the United States Patent and Trademark Office was presented with several computer implemented process claims. The imminent struggle to determine if software could fall within the scope of patent protection had begun\textsuperscript{26}.

\textsuperscript{23} For a detailed explanation of the process and components of executable software, please see R. PLOTKIN, “Computer Programming and the Automation of Invention: A Case for Software Patent Reform”, supra note 9.

\textsuperscript{24} Please see infra 65.

\textsuperscript{25} Software is entitled to patent protection. Nonetheless, several aesthetic aspects of computer software are still protected by copyright such as screen displays which can be protected as audiovisual or pictorial works.

Around 1968, the Court of Customs and Patent Appeals (CCPA) was faced with a series of appeals from the United States Patent and Trademark Board of Appeals in which the issue of the patentability of software related inventions was brought to question. In all cases the claims were initially rejected by being considered as non statutory in accordance with section 101 of Title 35 of the United States Code. The cases would induce the CCPA to carefully analyze the so called “mental steps” exception from 1945.

In this connection, the CCPA ruled that a useful process even when it could be implemented by mental steps would not make it ineligible for patent protection, firmly allowing the patentability of mental steps claims. Such a decision could be interpreted as an allowance to the protection of processes like algorithms by means of patents. Notwithstanding that in a rehearing procedure the Court stated that since the claims did not provide with an accurate description of the subject matter, patent protection could not be awarded, in such procedure the issue of patentable subject matter under section 101 was not modified.

Further technology developments lead the Court to determine that when a device was programmed innovatively and non obvious way, it would be physically different from the same machine without the innovative program, due to the fact that the elements composing its memory would be different.


27 Starting with In re Prater, 415 F.2d 1378, 159 U.S.P.Q. (BNA) 583 (C.C.P.A. 1968), the CCPA would continue to analyze patent cases in which computer software was involved.

28 Section 101 of Title 35 of the U.S.C. refers to patentable subject matter.

29 In re Heritage, 150 F.2d 554, 556, 66 U.S.P.Q. (BNA) 217, 220 (C.C.P.A. 1945) (holding that “purely mental acts are not proper subject matter for protection under the patent statutes”). Said doctrine is based on the following tenants (1) if a claimed method consists wholly of “mental” steps, then the subject matter is not patentable, (2) if a claimed method consists of both physical steps and mental steps, and the novel element lies in the mental steps, then the subject matter is not patentable, and (3) if a claimed method consists of both physical steps and mental steps, and the novel element lies in the physical steps, then the subject matter is patentable. Please see G. J. MAIER & R. C. MATTSON, “State Street Bank in the context of the Software Patent Saga”, supra note 26.


31 Such a decision was developed In re Bernhart n32  417 F.2d 1395, 163 U.S.P.Q. (BNA) 611 (C.C.P.A. 1969).

In re Mahony, the Court would continue to ascertain the patentability of computer programs. In this occasion the case was decided by ruling that the patent claims would be suitable in accordance with Section 101 notwithstanding the fact that such claims did not expressly refer to a machine or a not mental process. Furthermore, it would later be ruled that technical steps could become statutory subject matter processes under 101 so long as they are within the frame of technological arts, in accordance with the United States Constitutional mandate of promoting the progress of useful arts.

In 1972 the Court decided again not in favor of the Board of Appeals regarding a claim which encompassed a mental process, given to the fact that the claims could only be performed on hardware or would not have further use aside of with a computer. Nonetheless, the decision of the CCPA was further appealed before the Supreme Court of the United States, which in turn ruled that software was outside the scope of Section 101 since a patent for software would technically be to offer patentability for ideas. However, the Supreme Court also declared that such a decision should not be taken as the rejection of software patentability, and that the United States Congress intervention should settle the issue.

Eight years later, a seminal decision heavily based on the Committee Reports which followed the Patent Act of 1592 in which the Congress decided that everything under the sun made by man would be subject to patent protection broadened the scope of Section 101, forbidding only the patentability of laws of nature, natural phenomena and abstract ideas.

Concurrently the CCPA began developing procedures to determine the patentability of computer implemented inventions. The developments would be later materialized in the Freeman-Walter-Abele test, which had the specific purpose of analyzing if a proposed algorithm was being sought to be patented by itself or encompassed within physical means of the claim.

---

33 In re Mahony, 57 CCPA 939, 421 F.2d 742, 164 USPQ 572 (1970), the case involved a process for distinguishing data bits from framing bits in a bit stream.


35 Gottschalk v. Benson, 409 U.S. 63, 175 U.S.P.Q. (BNA) 673 (1972). This case involved an invention directed to the conversion of a binary coded decimal into a pure binary number.

36 The Freeman-Walter-Abele test was developed after a series of cases by which the CCPA tried to determine which software was entitled to patent protection. It was discarded almost two decades later by the United States Court of Appeals for the Federal Circuit. Please see G. J. MAIER & R. C. MATISON, “State Street Bank in the context of the Software Patent Saga”, supra note 26, see also J. E. COHEN & M. A. LEMLEY, “Patent Scope and Innovation in the Software Industry” California Law Review
During that time the Supreme Court was faced with Diamond v. Diehr\(^{37}\), and following the Chakrabarty decision the Court put special attention in determining if the claims sought an algorithm as an idea or if it was applied specifically to physical application and admonished that courts should not limit patent laws if such limitations where not established by the legislature\(^{38}\).

After a series of cases the Freeman-Walter-Abele test stated that “if the claim would be otherwise statutory, albeit inoperative or less useful without the algorithm, the claim likewise presents statutory subject matter when the algorithm is included”. However, the test was developed based on obsolete experiences and was discarded until In re Alappat\(^{39}\).

The USPTO began following the lead of the precedents set by the Supreme Court as well as the Federal Circuit, allowing the patentability of software. However, it was until State Street Bank & Trust Company v. Signature Financial Group, Inc.\(^{40}\), when the Court decided that a mathematical algorithm\(^{41}\), application or calculation, as long as it produces a useful and tangible result is then subject to patent protection\(^{42}\). This case was further reinforced by AT&T Corp. v. Excel Inc.\(^{43}\).


\(^{38}\) The Diehr decision was a signal for patent attorneys to bundle software inventions with hardware in order to obtain a patent, in other words, claiming hardware aiming at obtaining software protection. This would be later referred to as the “magic words doctrine”. Please see D. S. EVANS & LAYNE-FARRAR, “Software Patents and Open Source: The battle over Intellectual Property Rights”, supra note 15, J. E. COHEN & M. A. LEMLEY, “Patent Scope and Innovation in the Software Industry” supra note 36.

\(^{39}\) In re Alappat, 33 F.3d 1526, 31 U.S.P.Q.2d (BNA) 1545 (Fed. Cir. 1994). Aimed at “a means for creating a smooth waveform display in a digital oscilloscope”, the decision opened the door for software developers which would be entitled to claim software patents without claiming for something else. In this case the Court concluded that “a general purpose computer in effect becomes a special purpose computer once it is programmed to perform particular functions pursuant to instructions from program software.”


\(^{41}\) This case involved a patent covering a data processing system for implementing an investment structure which was developed for in Signature’s business as an administrator and accounting agent for mutual funds. In essence, the system, identified by the proprietary name Hub and Spoke, facilitates a structure where mutual funds (Spokes) pool their assets in an investment portfolio (Hub) organized as a partnership. Please see R. P. MERGES, P. S. MENELL & M. A. LEMLEY, “Intellectual Property in the New Technological Age”, supra note 1 at 1065 or G. J. MAIER & R. C. MATTSON, “State Street Bank in the context of the Software Patent Saga”, supra note 26.

\(^{42}\) “we hold that the transformation of data, representing discrete dollar amounts, by a machine through a series of mathematical calculations into a final share price, constitutes a practical
whereby the Court stated that as long as an invention containing an algorithm produces a tangible and useful result, it should be allowed patent protection.

The preceding paragraphs constitute a very short overview of the development of software patents in the United States. However it should be noted that the magic words doctrine is widely in use and its effects are still being developed in such countries.

3. The outcome and its consequences.

In accordance with the Intellectual Property Law of several countries outside the United States and Japan, patents for computer implemented inventions have been granted in a steadily growing basis, even in the European Union where not long ago a fierce debate regarding the patentability of software occurred. By means of the magic words doctrine, several patents which scope and intent is focused mainly in computer software are being granted by the mere addition of a technical implementation of such programs in order to circumvent the prohibition of patentability of software as such.

Nonetheless, it is important to comment that major drawbacks for such a policy arose in the U.S., which legal system followed the “magic words” doctrine until the Courts opened the way for the granting of pure software patents. Fortunately, the trial an error followed by a country in this case may help other countries to implement prophylactic policies in order to avoid the consequences faced by the pioneers in the field of software patents.

application of a mathematical algorithm, formula, or calculation, because it produces “a useful, concrete and tangible result”—a final share price momentarily fixed for recording or reporting purposes and even accepted and relied upon by regulatory authorities and in subsequent trades.”

43 AT&T Corp. v. Excel Communications, Inc., 172 F.3d 1352, 50 U.S.P.Q.2d (BNA) 1447 (Fed. Cir. 1999), cert. denied 120 S. Ct. 368 (1999). The Court concluded that as long as the software generated a “useful, concrete and tangible result”, patent protection should be granted, specifically rejecting that the claim should have a physical structure associated with for such protection.

44 In Mexico, Brazil or the EU.


46 Please see supra 38.

Notwithstanding the above, some of those mistakes are already taking place within the legal frame of Patent Offices granting patents for computer implemented inventions. For example; without an adequate classification system which allows a trustworthy search of inventions composed of computer programs\(^{48}\), the development of the IP frame of those countries in order to accept software as rightfully patentable invention will be deterred since the prior art databases will not be helpful enough to determine the novelty or non-obviousness of the software applications.

In light of the above, the following issues must be dealt with for anticipating the development of the IP frame in the field of software protection.

3.1 Major obstacles

As the granting of software patents is the legal trend that most Patent Offices will certainly follow in the evolution of national patent law\(^{49}\), steps should be taken to diminish the negative effects of uncertainty in the field of software prior art, which seems to be one of the major setbacks of software patents in the United States\(^{50}\).

At this stage the IP Offices of several countries are steadily granting patents for computer implemented inventions and classifying such inventions in their corresponding fields of art. Such practice is definitively in line with the patent legislation of said countries but will direct them to major complications when the examination of pure software patents becomes available, essentially causing the following issues:

a) The first and most important setback will be suffered by society, since inventors will face the risk of wasting valuable time and resources in patent searches with high probabilities of pursuing the development of a solution already existing in the prior art, along with the consequential risks of an infringement suit. As a result, the purpose of the patent system will be severely diminished in the field of computer software as incentives to innovate will be diluted.

\(^{48}\) It should also be stated that the omission to classify computer implemented inventions in a special software category is in compliance with the patent law of a country which does not grant patent protection for software as such.


Such environment will attract a lot of uncertainty in the field of computer software research and development, possibly depriving the consumers of new and more efficient products\textsuperscript{51}. Likewise, entrepreneurs wishing to obtain benefits from the knowledge pool constituted by the already available prior art will not be able to access its full benefits.

\textbf{b)} On the other hand, the examiners will not have accurate access to prior art while studying an application which should not be granted on the fields of novelty or inventive step, running the high risk of granting void patents which firstly will affect the society’s trust in the patent system. Likewise, companies will not have accurate information if faced with questions of enforcement or IP valuation while trying to access capitals. Startup companies may be bullied out or forced to pay royalties for weak patents, mostly due to the fact that royalties could be cheaper than a infringement suit or challenging the validity of the patent.

Additionally, it should be taken into account that Patent Offices will be pushed by new technology trends to hire skilled examiners in the field of computer software in order to properly deal with patent applications on such specialized field, since even only accurate access to prior art will not cover the necessary expertise to understand the claims and determine if the possible applications are indeed subject of patent protection.

\textbf{c)} The phenomena of submarine patents\textsuperscript{52} will likely occur when the impossibility of accurately determining what is the current state of the art will lead companies to develop a new technology which may infringe a patent lost in the prior art. The resulting infringement suit may become highly expensive to the company, and even the idea of a settlement could severely limit the possibilities of engaging in further research and development of new products\textsuperscript{53}.

\textbf{d)} Standardization will also become very complicated to achieve since main bodies like the International Telecommunication Union (ITU), which work closely with patent pools for establishing the best technology available as a standard could be confronted by patents which were submerged in the prior art held by a company which is not a part of the patent pool and


\textsuperscript{53} Please see A. E. ANDERSON, “Taming The Code: Effectively Implementing Software Patents”, supra note 50.
which claims cover a part of the standardized technology, dealing a severe blow to the standardization effort\textsuperscript{54}.

The overall result for the consequences of failing to duly classify the software patents since the early stages of legal development will surely take its toll as software developments arise.

e) Finally, it is very important to implement accurate legislation allowing the reverse engineering of computer programs, not only aimed at achieving interoperability, but also due to the incremental nature of software development, which may require the reverse engineering of a software program in order to determine if a specific part of the code which is required to be used is on the public domain or if another road should be pursued.

The reverse engineering exception is highly important since particular techniques may involve the reproduction of the software which is being reverse engineered, and such unauthorized reproduction of a patent protected invention may constitute infringement, even when such reproduction is not directly intended to compete with the patent holder in the same market\textsuperscript{55}.

On the other hand, reverse engineering may be the only available way for a patent holder to obtain evidence of the infringement of a competing product. However, if reverse engineering is deemed as infringement in itself, obtaining evidence may prove almost impossible.

3.2 Possible solutions

The issues mentioned in the preceding section underline some of the most recurrent setbacks for the patentability of software as such, however, developing a new class for pure software patents would be highly advisable for Patent Offices apart from Japan or the U.S., notwithstanding the actual state of their national patent legislation. As countries will adopt the trend of software patents, cross-classifying the computer implemented inventions between their actual class and the new software class will greatly simplify the development of consistent and accurate prior art in the future.

Evidently, it would also be necessary for the Patent Offices to dig and reclassify the computer implemented inventions which have been granted in the past.

\textsuperscript{54} Please see G. A. STOOLS “Software Patents” supra note 9 at 110.

\textsuperscript{55} Please see J. E. COHEN & M. A. LEMLEY, “Patent Scope and Innovation in the Software Industry”, supra note 34
However, such a task would be most complicated due to the complexity, size and time frame necessary for performing the update mentioned above, and even for a Patent Office as well funded as the EPO the task would be incredibly complex, certainly almost impossible for less fortunate Offices throughout the world.\textsuperscript{56}

**Updating the prior art**

In this regard, three strategies should be implemented concurrently: the first should be to recover the computer software related prior art which is currently located inside the patent databases. As mentioned above, the task is overwhelming in almost every aspect of it, however, a viable option would be to use the resources which are actually at hand.

Since the patent examiners are in direct contact with prior art searches and patent databases, reviewing the actual state of the art in order to determine the novelty or obviousness of a patent application, said examiners are already trained to understand how prior art searches work and thus, would be in the best position to flag patents for computer implemented inventions which are already within the patent databases, uploading them into the new classification while focusing on the application examination workload. A program of such characteristics would have to be implemented prior to assigning personnel to sweep as thoroughly as possible the patent databases for updating the prior art.

The second and evident strategy would be to pinpoint new patents for computer implemented inventions also as software patents. In this regard and using the resources already available should not pose a serious threat to Patent Offices.

As a third strategy, it is important to note that prior art is not always available at a patent database. The development of computer software and its protection has evolved differently since its inception, sometimes protected by means of contractual relations, trade secret law or copyright; sometimes not protected at all. In consequence, a very important portion of prior art is scattered outside patent databases and generally out of the examiner’s attention.

However, collecting prior art data outside patent databases would become a second titanic goal in order to accurately develop a complete and concise software prior art database, as the references may be in books, magazines or software dispersed all over the internet for example. How to attack this issue?

It should be taken into account that the United States faced the same problem at the dawn of the actual software patent policy, in fact, a strong argument against

\textsuperscript{56} However, such situation could be solved with assistance agreements and access to international patent databases.
software patents was that the USPTO would not be up to the challenge of accurately examining software patents due to the lack of prior art.

Nonetheless, although it would be only natural for a newly developed area that the prior art would be scarce or not within the Examiner’s grasp, it would also be natural that after years of patent examination such prior art would grow and become more reliable. Nevertheless, the absence of prior art in a new field or the difficulties to obtain reliable databases are not a reasonable argument against granting patent rights to new inventions.

**Peer 2 Peer examination and submissions**

A viable option would be again to use already available resources. In this case the software developing community may become extremely useful for the development of a healthy patent system.

It is fairly reasonable that patent examiners from countries which are not granting pure software patents do not have developed the necessary skills to provide good quality examination in such a very technical field. However, software developing communities are a common in almost every country and in the field of software patents examination, could become a very efficient source of prior art and examination information.

Evidently, such option should not be taken lightly due to the nature of the project and the value of the information at stake, as risks of incorrect information or submissions made in accordance with particular interests, possible even from competitors should be carefully measured and timely stopped.

Such endeavor should be carefully implemented and tailored to the needs and realities of each country, directed carefully by the IP Offices and expert computer programmers along with the participation of the developers in the private sector, including the following essential characteristics.

1) Disclosure of sufficient information for determining the scope of the patent application, which do not necessary means the disclosure of the source or object code.

It is very important to highlight that said characteristic has to be finely tuned in accordance with the needs of the application holder, since such a disclosure could become potentially catastrophic if essential information is made available to competitors.

Additionally, the information should be made available for a limited period of time, after which the examiner will not accept further submissions. The challenge of an
application outside the submission period should be prosecuted under common opposition proceedings.

2) Availability of a submission system which allows sufficient description of the subject prior art sample as well as attachment submissions of supporting information and or documentation.

The purpose of this system will be to assist the examiner in the search of relevant prior art which may help to determine the scope of a software patent application. In consequence, prior art submissions should be accompanied by bibliographic information, samples or other pertinent data in order to accurately determine the validity of such submission and its possible inclusion in an official action.

Such a proposal will have the benefit of additional examination of software patent applications and will have the incentive of allowing the users of such system to actively pursue the rejection of patents which would be considered as not novel or obvious. Fine tuning rules for such a proposal should be made by patent offices in accordance with the resources available and particular needs.

It should be highlighted that peer to peer reviews and assistance have been studied and implemented in other fields with outstanding results. For example, the folding@home distributed computing project developed by the Stanford University Medical Center and sponsored by top private companies has set a network of shared processing power provided by personal computers and even playstation 3 systems to analyze the folding of proteins with the goal of understanding and finding a cure to serious diseases.

While the folding@home project illustrates only the huge amount of computing process that could be set up in a network, the NASA Clickworkers project reveals the power behind people putting themselves together for a single project which is more complex than only lending processing capacity. This particular example reveals how the websurfing community dramatically accelerated the process of mapping and calculating the age of the Mars battered surface by analyzing its

57 Please see http://folding.stanford.edu/.

58 The folding@home project analyzes the way proteins fold (taking a particular shape to serve a particular function) in order to understand the “misfolding” process by means of which proteins do not take shape properly causing diseases such as Alzheimer and thus, finding a cure. The folding process of a protein takes roughly a million of a second, while a computer can simulate a nanosecond in full 24 hours and thus, results of the folding would be expected in 30 years. However, the network which has been set up with approximately 100,000 processors speeds up the process allowing for quicker folding simulations and further studies. For more detailed information please visit the website of the folding@home distributed computing project.

craters between November 2000 and September 2001, performing accurately what a seasoned expert crater rater would have done in several months60.

The final and best example of an endeavor of such dimensions has been set up again by the United States with the development of the Peer to Patent Project, which was implemented by the USPTO with the consent of its developer, The New York Law School Institute of Information Law and Policy61. This interesting project is already allowing the public to submit prior art samples and relevant information regarding software and computer architecture patent applications with the corresponding authorization of the application owners62. In exchange, the application holders receive an expedited examination which shortens the patent examination procedure63.

The project which provides assistance for the patent examination process in the field of computer arts also allows the public to become involved with the patent procedure, greatly increasing public participation and awareness, taking full benefit of the knowledge pool outside the PTO archives and conventional sources of information.

The benefits of implementing such a system in countries which already grant patents for computer implemented inventions would be reflected in an important development of prior art knowledge acquired for sources apart the patent databases, which in other circumstances would be very difficult to access and also will allow the examiners to improve their examination process skills, due to the availability of additional information.

The enlargement of the prior art databases will certainly diminish the ill consequences described above, while improved examination will also reduce the amount of petty patents issued by the PO.

60 The NASA Clickworkers project took into account the existence of several entries for the same crater image, using them to sort out a reasonable margin of error. Detailed information is also available at Y. BENKLER, “Coase’s Penguin, or, Linux and the nature of the Firm”, supra note 3.

61 Please see http://dotank.nyls.edu/communitypatent/ or http://www.uspto.gov/web/patents/peerpriorartpilot/.


Evidently, one major drawback of such proposal would be the abuse of the system. However, the incentives for the software programming community, deeply interested in avoiding the issuance of petty patents may secure a smooth process. On the other side, software development companies will have the incentive of secure research and development knowing that the possibilities of competitors appearing with petty patents will decrease.

The system could be implemented with relatively low costs and could aide enormously in the issuance of high quality patents, its benefits outlasting the funds which would be used in such a project. Long term costs would also be concisely low as the operation of such a system would not be complicated for a reasonably skilled webmaster. Furthermore, it is necessary to stress that such an endeavor would become fully operational if its access is free.

Finally, such a process should not in any case replace the traditional opposition proceedings.

On the other hand, Patent Offices should also require investing significant funds in hiring patent examiners with a strong computer programming background. It should be noted that the area of software development moves at incredibly high speeds in almost all fields of industry. For such reason is not possible to initiate the examination of software patents without solid computer programming experience and examination practice, notwithstanding the experience benefits that will derive from the implementation of the system mentioned above.

**Further proceedings**

Regarding the patent substantive examination, in order to maintain a healthy system in general and in particular to improve the quality of the issued patents, countries should provide adequate opposition proceedings, as well as re-issuance and re-examination proceedings.

The implementation or simplification of opposition systems by means of which third parties are entitled to bring to the attention of the examiner relevant prior art which may lead to the refusal of the issuance of patent rights, as well as an efficient system of re-issuance and re-examination will greatly aid other countries to maintain a healthy patent system.

In this regard, allowing third parties to oppose the issuance of a patent that may be deemed as invalid will help to reduce litigation costs, which are mainly the reason why if a treat is made with a weak patent a company would be more willing to settle and pay for royalties rather than engaging in a costly suit against such patent. Such system will enhance the cost/benefit of the tradeoff between the society and an inventor for the granting of such an exclusive right.
It is nonetheless important to highlight that several countries already provide in their legislation means for proceeding with a patent opposition. However, not every country provides such option and thus, it would be in their best interest to proceed accordingly.

**Patented software or patent pending**

On the other hand, special attention should be brought to the marking of patented products as a mandatory obligation on behalf of the patent owner\(^\text{64}\). Enacting such a provision will definitively help software developers and programmers to determine with a certain amount of accuracy which patents are protecting a specific program and will facilitate the search for a particularly patented technology.

By proceeding as mentioned above, third parties interested in developing a computer application will be able to determine which patents are covering competing applications and thus, may be able to determine with better accuracy if proceeding with the development of such application would be appropriate or if by the risk of incurring in patent infringement would be better to focus on another field.

Such provision will not only provide advantages for developers, since investors will have further incentives to provide capitals to IT companies in a more friendly development environment where risks can be determined in advance and certainty is more feasible.

Examples of such practice can be found in the Adobe™ Reader™ software, which briefly displays at the startup page the patents and patent applications by which such program is protected\(^\text{65}\). Furthermore, by accessing to the help menu a full list of the patent and patent applications which protect the software can be found, along with a list of licenses and cooperative development projects which are implemented in said program.

**Standardization**

The software developing industry has recognized the relevance of setting standards and its benefits regarding compatibility, safety and interoperability

---

\(^\text{64}\) Please see A. E. ANDERSON, "Taming The Code: Effectively Implementing Software Patents", supra note 50.

\(^\text{65}\) Such a procedure would ease the lack of information faced by software developers, as mentioned by A. G. GONZALEZ, "The Software Patent Debate", supra note 50, or Please see F. PIERA, "IPR Protection of Computer Programs and Computer Software in the Global Market", supra note 52.
among others. Nonetheless, with the world wide web as the biggest example of standardization, several players on the field fear that software patents may hinder important standardization efforts, with the possibility of reaching an agreement regarding a standard, implementing it with all the corresponding investment and all of the sudden, find that the industry was infringing on a third party’s intellectual property right.

This issue has been dealt with by standardization bodies such as the World Wide Web Consortium (W3C) or the International Telecommunication Union (ITU) in their respective directives for implementing patent pools. Nonetheless, the possibility of one company putting its feet on the throat of a standard by means of intellectual property rights is always open.

Against such a possibility, the W3C for example manages certain disclosure requirements for its members in accordance with its Patent Policy aimed at obtaining a Royalty-Free license agreement in the event that patents arise when discussing a possible standard. If such a license is not feasible, a license conveying royalties could be achieved under Reasonable And Non-Discriminatory (RAND) terms, in line with the ITU provisions.

Notwithstanding the above, the possibility of IPR’s locking standardization efforts is always latent. On the other hand and as recognized by the ITU in its IPR Copyright Guidelines, copyright is not precisely the best option available for software protection, due to the fact that software protects the expression of an idea and thus, leaves open the possibility of competitors working around the copyright protected software in order to develop a competing product.

In a nutshell, patent rights regarding computer sciences are not a threat against standardization efforts. Patents as any other IP right is a tool which can be used to foster innovation and economical growth if used properly and within the boundaries set by law and a little bit of common sense, otherwise, it could lead to technology deadlocks and stifle innovation.

66 Please see G. A. STOOBS “Software Patents” supra note 9 at 109.

67 Among which we can find the American Society for Testing and Materials (ASTM) or the International Organization for Standardization (ISO).

68 Please see http://www.w3.org/.

69 Please see http://www.itu.int/net/home/index.aspx.

70 Please see http://www.w3.org/Consortium/Patent-Policy-20040205/.

71 It is worth mentioning that W3C withdrew the RAND provision of its Patent Policy of 2004.

72 Please see http://www.itu.int/ITU-T/ipr/.
Interoperability

Disclosure of the program code is very limited in the United States by Court standards. Consequently, the capacity of software developers to accurately determine if a proposed invention is already covered by a patent or even if they could “couple” their invention with a widely accepted program and thus, take advantage of the consequent network effects generated by it is quite limited.

However, it should be noted that a disclosure policy regarding software patents could not be suggested or implemented without further studies regarding the particular legislation of each country. Nonetheless, policies regarding the promotion of interoperability and reverse engineering should be also implemented in order to assure a healthy patent system.

Reverse engineering in particular should be allowed in order to study, isolate and if necessary, duplicate the unprotected elements of a particular software program. It should be noted that most patent legislations do not allow reverse engineering exceptions, and due to the nature of software, reverse engineering a patented computer program should almost necessarily involve an unauthorized reproduction of it and as a result, patent infringement.

4.- Conclusion


74 This term should be interpreted for the sake of clarity as coupling in the sense of interoperability.


76 Apart from some well sounded cases (European Commission v. Microsoft Corp.) it is apparent that application development firms take advantage of disclosing interfaces to both competing and complementary products. Customer satisfaction regarding interoperability and the associated network effects are highly regarded. Please see K. BLIND, J. EDLER & M. FRIEDEWALD, “Software Patents: Economic Impacts and Policy Implications”, supra note 6.

77 Please see J. E. COHEN & M. A. LEMLEY, “Patent Scope and Innovation in the Software Industry”, supra note 34.

78 If the method used for the reverse engineering procedure is decompilation, it may possibly involve an unauthorized reproduction of the patented software in the random access memory.
Although up to date there is not a single conclusive study unveiling the benefits or damages of implementing software patents, the empirical evidence drawn by the success in the field of software developing companies in the United States and the profits generated by such industry should be carefully taken into account\textsuperscript{79}.

However, the transition to a more open and consistent patent law in the field of computer science cannot be made without fine tuning the system of each country or region, taking into account the economical resources at hand and the recommendations stated herein. In any event, the old discussion regarding the patentability of software should be left aside with a focus on adequately implementing recommendations for a better development of prior art, technical staff and strong patents.

The statistics and reports of the success of the software industry should not be treated lightly, and while software is a complex technology, its development is not as complicated as in the biotech field. In consequence, the promotion of software technology, education and protection may be a powerful economic incentive for other countries in the field of SMEs. It is just a matter of starting where the last country left off.

\textsuperscript{79} Outstanding data regarding the success of the software developing industry in the United States can be found at R. J. MANN, "Do Patents Facilitate Financing in the Software Industry?", supra note 3. While other authors claim that such a success is not related to the patentability of software, A. G. GONZALEZ, "The Software Patent Debate", supra note 50, or Please see F. PIERA, "IPR Protection of Computer Programs and Computer Software in the Global Market", supra note 52, it is also important to note that studies to support such a conclusion do not exist up to date. Nonetheless and until such studies are developed, countries which do not grant software patents should carefully consider the U.S. statistics.