BILSKI V. KAPPOS [130 S.CT. 3218 (2010)] (A CRITIQUE FROM THE PERSPECTIVE OF ITS IMPACT ON SOFTWARE PATENTS)

A.S. Vishwajith and Nehaa Chaudhari[1]

INTRODUCTION

Scientific Development and Open Source

The concept of scientific development has always been associated with the free and open dissemination of knowledge. Consequently, the idea of patenting when it originated in the 20th century was regarded as unethical by a large section of academicians[2], since a patent is an exclusive property right that is limited by time. Further, the utilitarian justification on the grounds of encouragement of innovation and development of inventions for the greater good of society has been rejected by eminent economists, including Joan Robinson, for presenting an inherent contradiction and consequently a negative result.[3]

Open Source has been regarded by many as an alternative approach to the patent regime, and is said to have prevailed in the fields of biotechnology[4] and information technology in the pre-1980s era.[5] However, subsequently, due to pressures from economic and legal circles, the US Congress adopted a series of pro patent legislations, the most important one being the *Bayh-Dole Act*,[6] with other nations following suit with similar legislations and varying degrees of success.[7] However, after the Second World

[1] Students, Fourth Year, B.A. LL.B. (Hons.), NALSAR University of Law, Hyderabad.

[2] Charles Weiner, 'Patenting and Academic Research: Historical Case Studies' 12 Sci, Tech & Human Values 50-51(1987) 50.

[3] Joan Robinson, *The Accumulation of Capital*, (3rd edn 1956) 87

[4] A K Rai, 'Regulating Scientific Research: Intellectual Property Rights and the Norms of Science', 94 NW UL REV 77, 88(1999).

[5] John Willinsky, 'The Unacknowledged Convergence of Open Source, Open Access and Open Science' available at <http://www.firstmonday.org/issues/issue10_8/willinsky> accessed 01 September 2011.

[6] 35 USC 200-212(2000).

[7] Aldo Geuna & Lionel Nesta, 'University Patenting and its Effects on Academic Research: The Emerging European Evidence' 35 RES POL'Y 772,794-97(2006).

War, there was an emergence of a programming community that favoured the idea of open source. The first open source project in the field of information technology was formed in 1984, leading to the formation of the Free Software Foundation created by Richard Stallman, which worked on the ideas of a software toolbox (GNU) and a general public license (GPL)[8] Further, in 1997, the Open Source Definition was drafted by Bruce Perens as an alternative to the GNU/GPL approach of Stallman. Later, in 1998, Bruce Perens, Eric Raymond and others formed the Open Source Initiative, an organization that acted as a certification body for open source licenses.

In recent times, the war between the users of Free and Open Source Software (FOSS) and the rest has only intensified. Further, jurisprudence in the United States, is seen by many to be pro-patents in relation to software patents. This has been largely judge-made, and therefore, fraught with inconsistencies. In such a scenario, the case of *Bilski v Kappos*[9] *(Bilski Case)*, to be heard by the Supreme Court of the United States was seen by many as an opportunity to bring about some clarity on the law in this regard. The decision has been received with relief by patent holders, on the grounds that it expands the scope of patentability as opposed to restricting it, and, consequently, with disappointment by the supporters of the free software movement. [10]

Research Outline

Over the course of this paper, the argument of the authors is that while *prima facie*, the decision of the Supreme Court may appear to favour software patents, a closer analysis would reveal that the interpretation could, in fact, serve as a tool to restrict software patents from being issued. In furtherance of the same, the law relating to the patentability of software has been traced and analysed, and a critique of the judgment in the *Bilski Case* has been presented, with a special focus on the impact of the judgment in determining the validity of the Beauregard test and its applicability to software patents. In conclusion, the authors suggest adopting the Mischief Rule of statutory interpretation and present an alternative to the present patent regime.

Development of the Law Relating to Software Patents

Prima facie, software is not patentable material. Algorithms are specifically excluded under patent law, and therefore, the legal basis for software stemmed in 1981, from the

[8] Bruce Perents, 'The Open Source Definition in Open Sources' in Chris DiBona et al eds, *Voices From the Open Source Revolution* (1999).

[9] 130 S Ct 3218 (2010).

[10] Dana Blankenhorn, '*Bilski* Decision Affects only Bilski' <http://www.zdnet.com/blog/open-source/bilski-decision-affects-only-bilski/6764> accessed 10 September 2011.

case of *Diamond v Diehr*.[11] Here, the Supreme Court had held that it was permissible to patent a physical process that was controlled by software[12]. This decision had faced strong criticism in the legal circles and the Supreme Court was presented with an opportunity to re-visit it, for the first time in twenty years, in the *Bilksi Case*, at a time, when the stakes were higher, with over $11.4 billion being spent on patent litigation, costs far exceeding any profits made from licensing, and many advancements having taken place in the realm of information technology.[13]

Post 1982, most decisions in this area were from the Court of Appeals for the Federal Circuit.[14] In 1998, the court in question emerged with a decision in the case of *State Street Bank v Signature Financial Group*[15], where the scope of patents was expanded and the test of practical application and a useful, concrete and tangible result was laid down. If these were applied to any algorithm, calculation or formula, the court held, the same could be patented. (It is to be noted that this case may not be considered to be good law anymore, since the same has been overturned, if not explicitly overruled, in the *Bilski Case*.)

Subsequent to this, there were numerous other decisions where the enabling requirements for software patents were relaxed. Coupled with this, courts were more than willing to grant injunctions and award substantially large sums of money as damages, thus acting as a deterrent to challenges to patents, and increasing the standard for any challenge. As a direct consequence of this prevalent climate, it was profitable to obtain and maintain a patent, and the ease in obtaining a patent was directly proportional to the size of the firm.[16] This proved to be a death knell for the software industry, which had witnessed unprecedented growth and innovation, largely as a result of the absence of patents. With patents now being granted so freely, and emerging resultant concepts such as cross-licenses and licensing fees, potential competitors to large corporations have effectively been rendered redundant.

FOSS emerged as a response to this system, and proved to be a direct threat to existing software companies. The idea of a 'protected commons' and the system of the General

[11] *Diamond v Diehr* (1981) 450 U S 175.

[12] Ryan Paul, 'SCOTUS to hear *Bilski* Case, May be huge for software patents' <http://arstechnica.com/tech-policy/news/2009/06/scotus-to-hear-bilski-may-be-huge-for-software-patents.ars> accessed 09 September, 2011

[13] ibid.

[14] ibid.

[15] 149 F 3d 1368.

[16] Mishi Choudhary, 'Looking out for Bilski: Software Patents v FOSS' < http://opensource.com/law/10/2/looking-out-bilski-software-patents-v-foss> accessed 09 September 2011.

Public License prompted companies to cooperate and exchange ideas, since they were secure in the knowledge that these ideas would not be appropriated by others claiming rights over them. However, the menace was furthered by 'patent trolls' who sued parties refusing to pay for a license and FOSS companies were unable to compete with them. The willingness of companies to pay for license fees was exploited, since companies aimed to avoid the high costs of litigation.

ANALYSING THE IMPACT OF THE *BILSKI CASE* ON SOFTWARE PATENTS

The Patent System and FOSS Users

The patent system, poses two fundamental threats to FOSS users. First[17] , that the software itself would be eliminated as a result of patented software, which essentially creates a monopoly and claims over basic techniques and common features of many computer programs. Second[18] , that as a result of patents on "business methods" which are computer enabled, the use of computers, for carrying out business functions, (that traditionally employed other methods) will be monopolized.

Interpretations of the *Bilski Case*

While the Supreme Court had been preparing to hear the *Bilksi Case*, there was palatable excitement[19] amongst the supporters of the free software movement, as this was believed to be an opportunity where software patents were eliminated altogether. With the ruling having emerged, the members of the free software movement seemed to have been left wanting. The prevailing perception, not only amongst supporters of free software, but also legal practitioners specializing in patent law seems to be that the Supreme Court has failed to tackle the broader issue about the non patentability of thought and thought processes. The only significant aspect that did emerge was perceived to be the strong opinion voiced by Justice Stevens, where he claimed that the decision of the majority would cause "mischief" and that all business method patents should be done away with.[20] Therefore, the onus now seems to be on the Congress to restrict the scope of patentable subject matter, due to the restraint exercised by the Court, where it refused to *"...read into the patent laws limitations and conditions which the legislature has not expressed."*[21]

[17] ibid.

[18] ibid.

[19] Ryan Paul (n 11).

[20] *Bilski v Kappos* [130 S.Ct. 3218 (2010)] (Stevens, J., concurring)

[21] ibid (Kennedy, J.).

Another possible interpretation that may prove to be a cause of concern for the supporters of free software, is that in refusing to read in limitations that were missing in the wordings of the legislation[22] , the court may have widened the scope of these patents by holding that the machine or transformation test was not the sole test, and that the Federal Circuit Court had erred in holding it to be so.[23] In this case, the court has confined the non patentability of a process to an 'abstract idea' and no further. Various concerns have been expressed in this regard, with some concerned that the decision neither eliminates a single pre-existing software patent, nor does it raise the bar and set higher standards for the future.[24] In fact, it explicitly declines to do so.[25]

While these concerns that are being expressed are legitimate, it is the argument of the authors that the *Bilski Case* may have, in fact, done more good than harm. Bilski's application for a patent on hedging risks in energy commodities trading was rejected by the U.S. Patent and Trademark Office and the Court of Appeals for the Federal Circuit, on the grounds that the patent claim did not include either the use of a 'specialised machine' or 'transformation of matter'[26] . This narrow reading was rejected by the Supreme Court and the machine or transformation test was held not to be the sole determinant. The application was instead rejected as being abstract. It is submitted that this act of rejection in itself would be beneficial to the free software movement.

The *Bilski Case* may be viewed as one where there was a denial of a software patent, and this could subsequently lead to the denial of other software patents. Since the court did not adopt the view that business methods were non patentable,[27] it rejected the application on the grounds of it being abstract and that math was not patentable.[28] Further, it affirmed that adding names and affixing math in a particular context did not convert the abstraction into a tangible and patentable process.

It is submitted by the authors that as per the case, the implementation of the hedging system proposed was to be done by means of computer software, despite the language of the claim not being explicit as regards this point. It is further submitted that the claim was not one involving simple mathematical calculations, but about extensive statistics that needed to be analysed in a complex manner, which would not be possible without computer software.

[22] ibid.

[23] ibid.

[24] Maureen O' Gara, '*[Updated]* Supremes 'Defang Bilski: Software Patents Safe' <http://opensource.sys-con.com/node/1447752> accessed 07 September, 2011.

[25] *Bilski* (n 26) (Kennedy, J., Stevens, J., concurring, Breyer, J., concurring)

[26] ibid, Slip Opinion, 1.

[27] *Bilski* (n 26) (Kennedy, J.) Slip Opinion, 12-15

[28] ibid, 16.

The appropriate test for determining the patentability of software inventions has been laid down in various Supreme Court and Federal Circuit cases[29] prior to the *Bilski Case.* While a mathematical algorithm/computer program would not, in itself, constitute patentable subject matter, the practical application of the same may be eligible for protection.[30] Another important development was the Beauregard test, where patents were granted to computer software claimed as an article of manufacture or a machine and the invention had to be claimed as being embodied on a 'computer readable medium'[31] Thus, it proves to be a loophole which may be used to get around the non patentability of pure software. The rationale behind not patenting software is that in its purest form, no matter what its function, a computer program is only a series of equations, that is, a representation of an algorithm, following the same fundamental concept. As an extension of the logic, since a computer onto which the program has been loaded is a machine, the same is patentable as per the law.

Disregarding the Beauregard Test

In the *Bilski Case*, the judges were unanimous in holding that any algorithm, whether for a set of named variables, or for a series of equations, is an abstraction by itself. It is therefore, un-patentable[32]. By way of analysis, it is noted by the authors that the patent application claimed by Bilksi was not for a simple mathematical algorithm, but for an extremely complex and statistical one. As has been stated by the authors earlier, it would be neither prudent nor reasonable to assume that the same could be executed without being coded into software that is then loaded onto a computer. What is of great significance is that despite the presence of the components of the aforesaid Beauregard test, the Court found the algorithm to be abstract. It may be argued that the court might not have applied the Beauregard test, since Bilski did not explicitly name a stock computer on which his algorithm was loaded. However, the view of the authors is that this cannot be construed as evidence of the fact that the Court would have granted him the patent had he named such a computer. On the contrary, the majority decision of the Court mentions the case of *Parker v Flook*[33] several times, to hold that an algorithm and *'insignificant post solution activity'* was still an abstract algorithm.

It is further submitted by the authors that the very foundation of the Beauregard test is the machine or the transformation rule, the application of which was heavily diminished

[29] *Diamond v Diehr* (1981) 450 U S 175; *Parker v Flook* (1978) 437 U S 584; *Gottschalk v Benson* 409 U S 63 (1972); *In re Beauregard* 53 F 3d 1583 (Fed Cir 1995).

[30] *Funk Bros Seed Co v Kalo Co* (1948) 333 U S 127, 130.

[31] *In re Beauregard* 53 F 3d 1583 (Fed Cir 1995).

[32] *Bilski* (n 26) (Kennedy, J.) 16.

[33] 437 U S 584.

by the Supreme Court in the instant case.[34] Another basis for the Beauregard test is the useful, concrete, and tangible rule[35] , which was not accepted by the Supreme Court in the *Bilski Case.*[36]

In conclusion, it may be stated thus- Bilski's application was for a software patent. The algorithm consisted of complex statistical calculations that could not be reasonably performed in any other way, even though he did not make explicit mention of a computer and a computer program. Therefore, the Court has decisively rejected an application for a software patent. Not just that, the Court has also severely handicapped the Beauregard test, by striking down one pillar and reading down another pillar of the test. It is submitted that this test was the basis on which software patents were granted, and that this loophole was created in order to get around the non patentability of computer programs since they were, in essence, just algorithms. It is the submission of the authors that the ruling of the Supreme Court in the *Bilski Case* places Beauregard test loophole[37] and its applicability to software and computer programs and their patentability, in a precarious position.

Conclusion- The 'Mischief Rule' and the Possible Solutions

In the eighteenth and nineteenth centuries, there was a need to encourage reform and innovation, especially in the American society, since this was the period when immigrants began to settle in America. These immigrants brought with them skills and ideas which needed to be rewarded. Very simply worded, this defined the need and the backdrop of the patent system. The past few decades have seen the unprecedented rise of two sectors, namely information technology[38] and biotechnology. As a result, the system of patenting is used almost exclusively by these sectors.[39] The manufacturing sector has moved on to using trade secrets and the related law in order to stay ahead of their competitors. The persisting irony is that these sectors use patent law for reasons which are mostly not relevant to its formation.[40]

[34] *Bilski* (n 38) 5-8.

[35] *State Street Bank & Trust Co v Signature Financial Group, Inc* 149 F 3d 1368, 1373 (1998); *AT&T Corp v Excel Communications* 172 F 3d 1352, 1357 (1999).

[36] ibid (n 40)

[37] *In re Beauregard* 53 F.3d 1583 (Fed Cir 1995).

[38] Larry Downes, 'The Bilski Case and the Future of Software Patents' <http://cyberlaw.stanford.edu/node/6351> accessed 05 September 2011.

[39] The Prior Art, 'Eben Moglen on Bilski, Software Patents and Big Pharma' <http://thepriorart.typepad.com/the_prior_art/2010/08/eben-moglen-on-bilski-software-patents-and-big-pharma.html> accessed 09 September 2011.

[40] ibid.

The authors submits that in the words of the Supreme Court of the United States, the intention of the Congress in enacting the Patent Law and using wide terms to formulate it, was that *ingenuity should receive liberal encouragement.*[41] It is further submitted by the authors that legislation must be interpreted in a manner which achieves the object of the Act,[42] a view that was laid down by Lord Coke in the *Heydon's Case.*[43] In the case of computer programs and software, a great amount of development is made by users who want to share and dispense their knowledge, rather than own it and it is these users that form the bedrock of the FOSS movement. For them, instead of proving to be a technique to maximise and incentivise innovation, the existing patent regime poses significant problems. In fact, in 1954, the Congress had admitted that when it spoke about the idea of patenting processes, it did not mean that the forces of nature and mathematics could be patented; it was, instead, talking about the application of the machine or transformation test.[44]

It may be viewed by way of argument that granting patents in the area of computer programs and software leads to the government granting monopolies, which if not curtailed, would continue to be used and exploited by the patent owners to the detriment of public interest. While patent law might have served the needs of the society in the preceding centuries, what is required now is a review of this law by way of a cost benefit analysis, an analysis which is unfailingly carried out for every other activity that involves government spending and expenditure, so that decisions may be educated and informed. It is submitted that under patent law, the underlying assumption is that granting a monopoly, albeit for a limited period, is an immeasurable benefit due to which it wields a great deal of influence over the public by means of regulation.

While the authors does not seek to criticize the patent system in its entirety, it is submitted that in the case of patents for software and computer programs, a review of the system, especially from the perspective of public interest, is a pressing need at the moment. Large corporations following the fundamental principles of capitalism, do not consider public interest to be important, since they are answerable only to their shareholders and it is in their interest to protect their property which they do through patents. It is the duty of the law makers and the courts to take into account and to protect public interest, a task which will not be achieved if this system continues to exist. Strict monopolies created out of the patent regime prevent free dissemination of information, which is definitely detrimental to public interest. This intersection between the patent system

[41] *Diamond v Chakrabarty* 447 U S 308-309.

[42] Vepa P Sarathi, *Interpretation of Statutes* (5th edn Eastern Book Company Lucknow 2010) 22.

[43] (1584) 3 Co Rep 7a

[44] *Bilski* (n 38).

and various discoveries and inventions of the information age is proving to be deeply problematic,especially one where litigation would be a dreadful solution to a persisting ache.

By way of a solution, with a view to reconciling the interests of both sides of the debate, the patentees and the users of FOSS, the authors propose that the possibility of granting patents for a shorter period, say between two to five years as opposed to twenty years may be explored. As an additional caveat, compulsory licensing at rates pre-determined by the government or at rates determined on a case to case basis may be imposed. While this suggestion needs further debate on its utility and applicability to the existing system, it might, if implemented and developed as a concept, prove to be a viable solution to the problem of creation of monopolies as a result of the present system of patenting. In the meantime, if it was not the intention of the Congress to grant patents in these areas, and if the effect of granting patents is, in effect, negating the object of the Act, it is time to reconsider whether computer programs and software should be patented at all.