

Patenting of Computer Related Inventions: A Look at *Bilsky*¹ and its Applicability in the Indian Scenario^{*}

Adithya Banavar

National Law School of India University,
Bangalore, India
adithyabanavar@nls.ac.in

Abstract. On 30th October 2008, the United States Court of Appeals for the Federal Circuit held en banc that a computer related invention is patentable if “(1) it is tied to a particular machine or apparatus, or (2) it transforms a particular article into a different state or thing”. The ‘useful, concrete and tangible result’ test laid down by the same Court in *State Street*² was effectively overruled. On June 1st 2009, the Supreme Court of the United States of America has granted a petition for a writ of certiorari against the decision in *Bilski*. *Bilski* is regressive in many senses, but is also consistent with precedent. The aim of this paper is to study the evolution of the law in relation to software patents over the years in the USA and compare this with the present Indian position. This paper aims to outline the position of law in the United States as it stands now. It also advances arguments as to what the United States Supreme Court should hold in the *Bilski* case, now that it has granted an order of certiorari. It then looks at the position of Indian law before concluding on how the Courts can interpret the present law and whether any change in the law is needed to that effect. In India, the statute excludes from patentable subject matter – “computer programs per se or algorithms”. There is no Indian case law in relation to the interpretation of this clause. Clearly, how this is interpreted would make a considerable difference to the patentability of software in the Indian scenario. The paper is limited in scope to legal arguments and does not look at policy questions, as to whether software should be granted patent protection or not.

1. History of Patentability of Software in the USA

The issue that arises here is the interpretation of the word ‘process’ that appears in the United States Code in context of patents. 35 U.S.C. § 101 reads “Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefore, subject to the conditions and requirements of this title.” 5 U.S.C. § 100(b) defines ‘process’ as meaning “process, art or method, and includes a new use of a known process, machine, manufacture, composition of matter, or material.” This definition is most unhelpful, especially given that in defining process, it uses the word process. In this Section, the journey of software as patentable subject matter is seen with respect to the Supreme Court of the United States [“SCOTUS”], The Court of Customs and Patent Appeals [“CCPA”] and the Court of Appeals for the Federal Circuit [“CAFC”] that was created by Congress in 1982 primarily to hear patent cases.

1.1 Doctrines prior to *Benson*

1.1.1. *Mental Steps and Functions of Machine Doctrines*

Prior to the various SCOTUS decisions on the topic, the CCPA had taken the stand that processes involving mental operations were considered unpatentable. This was applied for any mathematical algorithm as well as any invention where the primary inventive component was a mathematical algorithm. This meant that no software process could possibly be patented.³ The ‘function of a machine’ doctrine is generally traced to a 19th Century

¹ *In Re Bilsky*, 545 F.3d 943.

^{*} This paper was originally published in Kierkegaard, A. (2009) *Legal Discourse in Cyberlaw and Trade*. IAITL-
² *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*, 149 F.3d 1368

³ See *In Re Heritage*, 32 C.C.P.A. (Pat.) 1170. See also *In Re Shao Wen Yuan*, 38 C.C.P.A. (Pat.) 967. The mental steps doctrine was based upon the familiar principle that a scientific concept or mere idea cannot be the subject of a valid patent. See *In Re Bolongaro*, 20 C.C.P.A. (Pat.) 845. C.f. *Diamond v. Diehr*, 450 U.S. 71 (1981).

judgment of the Supreme Court of the United States.⁴ The Court stated in that case - "it is well settled that a man cannot have a patent for the function or abstract effect of a machine, but only for the machine which produces it."⁵ By two judgements in 1968,⁶ *In Re Tarczy-Hornoch* and *In Re Prater*, the CCPA did away with the above two doctrines stating that they were a misinterpretation of precedent.

1.1.2. Technological Arts Doctrine

In 1970,⁷ the CCPA stated that any sequence of operational steps was a patentable process so long as it was within 'technological arts' so as to be in consonance with the Constitutional mandate to promote the progress of useful arts. Article I, Section 8 states "The United States Congress shall have Power...To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries."

The case dealt with the patentability of a method of improving the delineation of geological subsurface formations by taking a series of seismograms from geographically separated stations by using a hyperbolic function to the family of seismograms produced by a particular arrangement of stations. The claim was rejected as the novelty lay in the mental process as opposed to the physical steps. The CCPA however, reversed and rejected the point of novelty approach followed by the patent office. On the point of novelty approach, if the novelty or advancement in the art claimed by the inventor resided solely in a step of the process embodying a mental operation or other unpatentable element, the claim was rejected under § 101 as being directed to nonstatutory subject matter. The Court relied on *Prater* to hold that the mental steps doctrine had been done away with and hence, the 'point of novelty' approach also lost its footing.

The 'technological arts' standard was refined in *Re Benson*,⁸ in which the court held that computers, regardless of the uses to which they are put, are within the technological arts for purposes of §101.

*1.2. Gottschalk v. Benson*⁹

This case related to the patentability of an algorithm to convert binary-coded decimal numbers ["BCD"] into purely binary numbers. The Court relied on *Funk Bros. Seed Co. v. Kalo Co.*,¹⁰ where it was held that newly discovered phenomena of nature could not be patented and extended the logic from a product patent to a process patent and included mental processes or abstract intellectual concepts as not patentable, simply because "they are the basic tools of scientific and technological work". Keeping this in mind, the Court took the stand that the conversion of BCD was too abstract to be patentable as it covered both known and unknown uses of the algorithm. In the reversal of the CCPA judgment, the Court did not refer to the new 'technological arts' doctrine or to the rejection of the 'mental steps' and 'functions of a machine' doctrines.

This decision has often been interpreted as a prohibition of all software patents across the board. This is perhaps supported by the statistics that show that in the decade that followed this judgement, virtually no one applied for a software patent.¹¹ While the judgement may have given that impression on first reading, all that the judgment states is that the phenomena of nature cannot be patented. The following observation perhaps led to this conclusion - "The mathematical formula involved here has no substantial practical application except in connection with a digital computer, which means that if the judgment below is affirmed, the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself." The judgment also stated in conclusion that "[i]t may be that the patent laws should be extended to cover these programs, a policy matter to which we are not competent to speak. The President's Commission on the Patent System rejected the proposal that these programs be patentable." As explained in the judgment itself, the Commission report advocated the rejection of software patents as it believed that the patent office lacked a reliable classification as well as a searching technique and hence could not examine applications.

The number systems and conversion techniques have certain innate characteristics and properties that cannot be changed, but merely discovered. No patent is possible on such abstract concepts. If limited to the facts of the case, this is the possible understanding of the judgement. The judgement stated that "It is said that the

⁴ *Corning v. Burden*, 56 U. S. 252 (1854).

⁵ The doctrine had been reaffirmed in numerous other judgements of SCOTUS. See *Risdon Iron & Locomotive Works v. Medart*, 158 U. S. 68, (1895); *Westinghouse v. Boyden Power Brake Co.*, 170 U. S. 537 (1898); *Busch v. Jones*, 184 U. S. 598 (1902); *Expanded Metal Co. v. Bradford*, 214 U. S. 383, (1909).

⁶ See *In Re Tarczy-Hornoch*, 55 C.C.P.A. (Pat.) 1441, (1968) and *In Re Prater*, 56 C.C.P.A. (Pat.) 1381, (1968).

⁷ See *In Re Musgrave*, 57 C.C.P.A. (Pat.) 1352 (1970).

⁸ *In Re Benson*, 58 C.C.P.A. (Pat.) 1134, (1971).

⁹ *Gottschalk v. Benson*, 409 U. S. 63 (1972).

¹⁰ *Funk Bros. Seed Co. v. Kalo Co.*, 333 U.S. 127 (1948).

¹¹ G. A. Stobbs, *Software Patents*, 137 (New York: Aspen Law & Business, 2nd Edition, 2000).

decision precludes a patent for any program servicing a computer. We do not so hold.” However, if one was to look merely at the words of the opinion, it seems that the Supreme Court was indeed saying later that patents could not be obtained for computer programmes. “If these programs are to be patentable, considerable problems are raised which only committees of Congress can manage, for broad powers of investigation are needed, including hearings which canvass the wide variety of views which those operating in this field entertain. The technological problems tendered in the many briefs before us indicate to us that considered action by the Congress is needed.”

It is interesting to note that after looking through all the precedent on process patents, the Supreme Court rejected the argument that a process patent must either be “tied to a particular machine or apparatus or must operate to change articles or materials to a ‘different state or thing.’” The Court opined that they were not limiting the process patent only to these two conditions.

1.3. INTERPRETATION OF BENSON

1.3.1. Point of Novelty Approach

The CCPA resurrected this approach post *Benson* and held that a process claim in which the only novel part was a mathematical equation to be solved as the final step of the process was not patentable.¹² This approach was, however, short lived and the CCPA sought to narrow the interpretation of *Benson* with each new case that came before it.

1.3.2. Apparatus/Process Distinction

In 1974, the CCPA held that *Benson* applied only to process claims and not to apparatus claims. The case related to a recordkeeping system that involved a digital computer with a programme.¹³ In dissent, Judge Rich stated that limiting *Benson* to process claims would make the question of patentability turn on the form or means in which the program invention was claimed as opposed to the substance of the claim. In 1976, again, the CCPA upheld the earlier judgment and based its decision on the reasoning that a computer with a programme is structurally different from the same computer. Judge Rich found himself in dissent again, this time joined by Judge Lane, who both felt that *Benson* had barred the patent of computer programmes across the board. The case related to an apparatus for scanning and converting data signals. *Benson* was distinguished by stating that this case related to a specific purpose and operated in relation to a particular technology as opposed to a ‘phenomena of nature’.¹⁴

1.4. *Parker v. Flook*¹⁵

Six years after *Benson*, a question arose as to whether the “identification of a limited category of useful, though conventional, post-solution applications of such a formula” makes patentable a mathematical formula not otherwise patentable (as under *Benson*). The patent related to an algorithm that updated ‘alarm limits’. As explained in the opinion of the Court, “[d]uring catalytic conversion processes, operating conditions such as temperature, pressure, and flow rates are constantly monitored. When any of these ‘process variables’ exceeds a predetermined ‘alarm limit’, an alarm may signal the presence of an abnormal condition indicating either inefficiency or perhaps danger. Fixed alarm limits may be appropriate for a steady operation, but during transient operating situations, such as start-up, it may be necessary to ‘update’ the alarm limits periodically.”

The patent officer had rejected the application on the grounds that the difference between prior art and the invention was merely an algorithm and a patent would in practice be on the formula, and this was not patentable as under *Benson*. The CCPA reversed and held that since the mere solution of the algorithm would not be an infringement, a patent on the method would not pre-empt the formula. The acting Commissioner of Patents and trademarks filed the petition for the writ of certiorari as he felt that the number of patent applications would increase exponentially given the rapidly expanding software industry.

The Court held that the words in *Benson* relating to pre-empting the entire mathematical formula would not apply here, as there are numerous uses of this formula outside the petrochemical industry that would remain in the public domain. The Court reasoned that an assumption ought to be made that the principle or mathematical formula were well known, and the invention was merely a new mode of applying it. The Court derived this

¹² See *In Re Christensen*, 478 F.2d 1392 (1973).

¹³ See *In Re Johnston*, 502 F.2d 765 (1974).

¹⁴ See *In Re Noll*, 545 F.2d 141 (1976). SCOTUS did not grant certiorari against this judgement. See 434 U.S. 875 (1977). See also *In Re Chatfield*, 545 F.2d 152.

¹⁵ *Parker v. Flook*, 437 U.S. 584 (1978).

reasoning from *O'Reilly v. Morse*, 56 U.S. 62, which in turn had applied the reasoning of an English Judgement of the Court of the Exchequer, *Neilson v. Harford*, (1841) 151 ER 1266. The latter case dealt with a patent application for a discovery that a blast furnace was more efficient if the air was heated before being blown through the molten iron. His 'invention' related to passing the air through a heating chamber. The Court granted patent and drew a distinction between a principle and the application of a principle. In the words of Baron Parke, "We think the case must be considered as if, the principle being well known, the plaintiff had first invented a mode of applying it by a mechanical apparatus to furnaces; and his invention then consists in this — by interposing a receptacle for heated air between the blowing apparatus and the furnace."

Based on this reasoning, since the output of the formula was merely a number, the Court opined that this was akin to seeking a patent for 'discovering' that the formula for the calculation of a circumference of a circle can be used for calculation of circumference of a wheel. The Court as in *Benson* held that this decision did not lay down a blanket ban on software patents, but just that when stressing the distinction between a principle and its application, and considering that the principle was prior art, there was no new use discovered and that the discovery of phenomenon cannot be patented unless there is some other inventive concept in its application.

The Court held that "the youth of the industry may explain the complete absence of precedent supporting patentability. Neither the dearth of precedent, nor this decision, should therefore be interpreted as reflecting a judgment that patent protection of certain novel and useful computer programs will not promote the progress of science and the useful arts, or that such protection is undesirable as a matter of policy. Difficult questions of policy concerning the kinds of programs that may be appropriate for patent protection and the form and duration of such protection can be answered by Congress on the basis of current empirical data not equally available to this tribunal."

1.5. POST FLOOK PATENTABILITY OF SOFTWARE

1.5.1. Freeman-Walter-Abele Test

The test arose from three judgements of the CCPA.¹⁶ The test had two parts. First, to determine whether the claim recites an algorithm within the meaning of *Benson* and second, determining whether the algorithm is "applied in any manner to physical elements or process steps".¹⁷ The test was rejected in *Re Alappat* and this rejection has been held inadequate in *Bilski*.¹⁸

1.5.2. Rejection of Flook

The CCPA critiqued *Flook* and stated that *Flook* essentially jumbled statutory provisions that were conceptually unrelated.¹⁹ The CCPA moved on to distinguish *Flook* on facts and ruled that *Flook* turned on the way the patent claim was drafted. The CCPA took the position that if a claim disclosed an entire process as novel, it would be patentable, even if the only novel element were a computer program. *In Re Diehr*²⁰ construed *Flook* in this manner.

1.6. Diamond v. Diehr²¹

Decided in 1981, the case related to the patentability of a process for moulding raw, uncured synthetic rubber into cured precision products that includes in several of its steps the use of a mathematical formula and a programmed digital computer. The Court held by bare majority of 5 to 4, that "continuous measuring of the temperature inside the mould cavity, the feeding of this information to a digital computer which constantly recalculates the cure time, and the signalling by the computer to open the press are all new in the art" and could hence be patented.

The Court reasoned that what was sought to be patented here was not an algorithm or formula, but a process for curing of rubber. While this process involved the use of a mathematical equation, they did not seek to pre-empt the use of that equation. What this patent does in effect is only to foreclose the use of that particular equation in conjunction with the other steps in the process. It was here that the Court laid down the machine or transformation test. In the words of Chief Justice Rehnquist, "On the other hand, when a claim containing a mathematical formula implements or applies that formula in a structure or process which, when considered as a

¹⁶ See *In Re Freeman*, 573 F.2d 1237. See also *In Re Walter*, 618 F.2d 758. See also *In Re Abele*, 684 F.2d 902.

¹⁷ See *In Re Abele*, 684 F.2d 902.

¹⁸ *Infra* Note 33.

¹⁹ See *In Re Bergy*, 596 F.2d 952.

²⁰ *In Re Diehr*, 602 F.2d 982.

²¹ *Diamond v. Diehr*, 450 U.S. 75 (1981).

whole, is performing a function which the patent laws were designed to protect (e.g., transforming or reducing an article to a different state or thing), then the claim satisfies the requirements of §101.” This test has ostensibly been applied by *Bilski* and states that software can be patented if connected to a machine or if part of a process that brings about transformation.

Justice Stevens in dissent opined that the majority had misunderstood the patent application and in effect had ‘trivialised’ the holding in *Flook* and the principle that underlies *Benson* along with a settled line of precedent that had interpreted these judgments. However, he concedes that if the reading of the patent by the Majority were correct, then it would be patentable. He states that curing of rubber is a patentable process as is clear from the grant of patent to Charles Goodyear many years prior. He continues that what is sought to be patented here is very similar to what was sought to be patented in *Flook*. He states, “The essence of the claimed discovery in both cases was an algorithm that could be programmed on a digital computer. In *Flook*, the algorithm made use of multiple process variables; in this case, it makes use of only one. In *Flook*, the algorithm was expressed in a newly developed mathematical formula; in this case, the algorithm makes use of a well known mathematical formula. Manifestly, neither of these differences can explain today’s holding.”

He finally advocated a complete ban on the patentability of computer related inventions (software). He saw two problems with the Majority’s ruling. “First, the cases considering the patentability of program-related inventions do not establish rules that enable a conscientious patent lawyer to determine with a fair degree of accuracy which, if any, program-related inventions will be patentable. Second, the inclusion of the ambiguous concept of an “algorithm” within the “law of nature” category of unpatentable subject matter has given rise to the concern that almost any process might be so described, and therefore held unpatentable... I believe both concerns would be better addressed by (1) an unequivocal holding that no program-related invention is a patentable process under § 101 unless it makes a contribution to the art that is not dependent entirely on the utilization of a computer, and (2) an unequivocal explanation that the term ‘algorithm’ as used in this case, as in *Benson* and *Flook*, is synonymous with the term ‘computer program’.” However, he found himself in the dissent and SCOTUS had finally allowed the grant of a software patent and ostensibly overruled *Flook*.

1.7. Post Diehr Cases on patentability of software

1.7.1. *In Re Alappat*²²

The CAFC sat *en banc* in 1994 to decide on the patentability of a means of creating “a smooth waveform display in a digital oscilloscope”. The effect of the ruling was clear wherein it allowed the patent of any software even if it ran on a general-purpose computer. The result of this was that a data structure, that is a way of arranging data, was granted patent protection. The Court held that the “printed matter exception” applied only to printed lines of characters “intelligible only to the human mind”. Based on this, software (data structure) in the instant case is processed by a machine and hence would never come under the exception.²³ Any computer programme *per se* was therefore patentable so long as it was more than a mere mathematical equation. Relying on an earlier SCOTUS ruling that Congress intended the patentability of “anything under the Sun that is made by man”,²⁴ the Court narrowed the application of the trilogy of SCOTUS cases and stated that under the three cases, it was clear that the only three judicially created exclusions were “laws of nature, natural phenomena and abstract protection”. The Court stated, “A close analysis of *Diehr*, *Flook*, and *Benson* reveals that the Supreme Court never intended to create an overly broad, fourth category of subject matter excluded from Section 101. Rather, at the core of the Court’s analysis in each of these cases lies an attempt by the Court to explain a rather straightforward concept, namely, that certain types of mathematical subject matter, standing alone, represent nothing more than abstract ideas until reduced to some type of practical application, and thus that subject matter is not, in and of itself, entitled to patent protection.” The Court then relied on pre *Flook* judgements²⁵ to hold that a computer program structurally altered a computer and thus, a general-purpose computer with a computer program would become a special-purpose computer.

1.7.2. *State Street Bank and Trust v. Signature Financial Group Inc.*²⁶

In 1998, the CAFC laid down the ‘useful, concrete and tangible result test’. The case related to a claim entitled “Data Processing System for Hub and Spoke Financial Services Configuration.” It was a data processing system

²² *In Re Alappat*, 33 F.3d 1526.

²³ See *In Re Lowry*, 32 F.3d 1579.

²⁴ *Diamond v. Chakrabarty*, 447 U.S. 303.

²⁵ See *In Re Noll*, 545 F.2d 141 and *In Re Prater*, 56 C.C.P.A. (Pat.) 1381 (1968).

²⁶ *State Street Bank and Trust v. Signature Financial Group Inc.*, 149 F.3d 1368.

facilitating the spokes (mutual funds) to pool their assets in a hub (investment portfolio) in the form of a partnership. The case relied on the earlier decision in *Alappat* and stated that to be patentable, an algorithm must be applied in a 'useful' way. The Court rejected the *Freeman-Walter-Abele* test and stated that after *Diehr* and *Chakrabarty*, the test has no application. "[W]e 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 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 and reporting purposes and even accepted and relied upon by regulatory authorities and in subsequent trades."

The case also held that business methods were not excluded from patent protection as under the law. The Court held that the only exceptions have been found in Title 35 or the judge-made exceptions of abstract idea, laws of nature or phenomena of nature. In *A.T. & T. Corporation v. Excel Communications Inc.*, 172 F.3d 1352, the CAFC found that a method involving electronic switches and a telecommunication system was eligible to be patented. The Court stated "any step-by-step process, be it electronic, chemical, or mechanical, involves an 'algorithm' in the broad sense of the term."

1.7.3. *In Re Bilsky*²⁷

In the recent *en banc* judgement by the CAFC, the Court rejected both the tests laid down in *State Street* as well as the *Freeman-Walter-Abele* test. The case related to a claim of hedging commodity transactions. The Court denied *Bilski* a patent and set down a single test for the patentability of software or business methods (or of processes in general). In the words of the Court, "purported transformations or manipulations simply of public or private legal obligations or relationships, business risks, or other such abstractions cannot meet the test because they are not physical objects or substances, and they are not representative of physical objects or substances."

The Court held that a process would be patentable if it is tied to a particular machine or apparatus or if it transforms a particular article into a different state. The question as to whether a general-purpose computer would come under the first test was left open for future decision. In Relation to the second test, the Court qualified it by stating "the involvement of the machine or transformation in the claimed process must not merely be insignificant extra-solution activity."

A recent judgement (November 24th, 2008) of the Board of Patent Appeals and Interferences ["BPAI"], *Ex Parte Halligan*, B.P.A.I. Appeal 2008-1588 dealt with this issue at length. The BPAI noted that the Federal Circuit provided "some guidance [to deal with this problem] when it explained that the use of a specific machine must impose meaningful limits on the claim's scope to impart patent-eligibility." The claims dealt with "a method [to account for trade secrets] performed on a programmed computer." The claims' scope was not limited as it added "nothing more than a general purpose computer that has been programmed in an unspecified manner to implement the functional steps recited in the claims."²⁸

In Judge Newman's dissent, she felt that inventions that deal with particular processes like data handling or photonic technology would be granted protection and would therefore be a disincentive in these emerging fields of technology.²⁹ Judge Rader, in his dissent stated, "today's software transforms our lives without physical anchors" and that this test effectively "not only risks hobbling these advances, but precluding patent protection for tomorrow's technologies." Another problem with *Bilski* is that it rigidly applies the test it has laid down and makes it necessary as opposed to a sufficient condition.³⁰ However, the case itself seems to have made it easy to distinguish in future by stating that "we certainly do not rule out the possibility that this court may in the future refine or augment the test or how it is applied." It seems that *Bilski*, though a reasoned judgement lacks the foresight required in interpreting the law, especially when it comes to technology.³¹ It is, however, not likely to be overruled in SCOTUS. The next Section discusses how SCOTUS should see *Bilski* and how it should rule on this issue.

²⁷ *In Re Bilsky*, 545 F.3d 943.

²⁸ See Stefania Fusco, "Is in Re Bilsky a Déjà vu?", *Stanford Technology Law Review* (2009).

²⁹ See Generally B.J. McEniery, "The Federal Circuit in *Bilski*: The Machine or Transformation Test", 91 J. Pat. & Trademark Off. Soc'y 253 (2009).

³⁰ See R.A. Hulse and R.R. Sachs, "Making Sense of the Revived Machine-or-Transformation Test in *In Re Bilski*", *Intellectual Property & Technology Law Journal* (2009).

³¹ See M.A. Shimokaji and P.L. Gahagan, "Mind Over Matter: The *Bilski* Decision, Like Others before It, Reveals How Courts Have Frequently Kept Patent Law Lagging behind Technology", *Los Angeles Lawyer*, 36 (2009).

2. *Bilski* and *Scotus*

Upon reading the case carefully, and understanding the limits placed by the Court in *Bilski*, it seems that *Bilski* is in perfect consonance with the thinking that prevailed in SCOTUS during the 1970s as opposed to the broader interpretation given by the CAFC in preceding years.³² As far as this goes, it seems to be in consonance with Stevens' dissent in *Diehr*.

Bilski is an opportunity for SCOTUS to look at the *State Street* test. In the ten years that *State Street* held the field and the four years before that under *Alappat*, SCOTUS declined to interfere with the foresight of the CCPA. The CCPA made it easier to get software patents, in light of rapidly advancing technology.

It is therefore felt that SCOTUS should overrule *Bilski*. If there is to be a proscription on software patents, it must be from Congress and cannot be by judicial mandate. The *State Street* test seems to be the right test for software patents and should be upheld by SCOTUS.

Another issue as seen from the CCPA's interpretations of *Benson* related to distinguishing the case by making an apparatus/process distinction. The problem with this is that a clever patent attorney need only draft his software patent as an apparatus claim thereby circumventing the tests under § 101.³³ SCOTUS will have to expressly state that its holding would apply to pseudo process claims that are in the guise of an apparatus claim. However, the question then arises as to where the line ought to be drawn between a pseudo process claim and an actual apparatus claim. SCOTUS should lay down guidelines dealing with exactly how a patent examiner should determine this, by looking at the substance of the claim as opposed to its form.

3. Indian Position on Patentability of Software³⁴

Section 3(k) of The Patents Act, 1970 states that "a mathematical or business method or a computer program per se or algorithms" are not inventions within the meaning of the Act and hence, not patentable. There is considerable debate on the interpretation of *per se* as well as that of computer program. The question that arises is whether computer program refers to any software, or only to a programming language or platform on which software can be designed. If it represents the former, then there seems to be a proscription on patents for software, while if it means the latter, the bar is merely on a type of software, on which it is possible to build other software, in order to ensure that innovation is not stifled by the grant of patent. The Patents (Second Amendment) Bill, 1999 did not contain the words *per se*. This was added on the recommendation of the Joint Committee set up by Parliament. "With respect to the patenting of software inventions, the Joint Committee recommended the insertion of the phrase "per se" in Section 3(k) of the bill and clarified that the phrase was being inserted to address the patentability of inventions relating to computer programs that may include certain other things 'ancillary thereto' or 'developed thereon'. The Joint Committee further clarified that the intention of the bill was not to reject such inventions from grant of patents but only to distinguish such inventions from the underlying computer programs as such."³⁵ This distinction seems oddly artificial and would be an impossible line to draw. All software is capable of improvement and any patent would stifle innovation in relation to that particular software. The general impression is that the only way to get software patented would be by amendment to the existing law.³⁶

An interesting twist, however, is the draft patent manual.³⁷ This document has no statutory backing, but regulates the procedure a patent examiner has to follow. As per this manual, software is made patentable if it has *technical application*. The manual divides patent claims in relation to computer inventions into three categories. Method/Process, Apparatus/System and Computer program product. In Relation to the first category, the manual states that in order to distinguish it from software *per se*, a claim for software directed towards a technical process would be patentable and distinct from mere software. The manual states –

³² See M.J. Edwards and D. Steinberg, "The Implications of *Bilski*: Patentable Subject Matter in the United States", 49 IDEA 411 (2009).

³³ See O. Armon and E. Gardner, "New Restrictions on the Patentability of Process Claims: Looking Beyond *In Re Bilski*", *Journal of Internet Law* (2009).

³⁴ Indian cases normally make copious references to judgements from the US and the UK. For instance, all the judgements on the right to privacy are based almost entirely on cases from SCOTUS. See *Gobind v. State of M.P.*, 1975 CriLJ 1111. Although SCOTUS judgements are not binding on Indian Courts, they have considerable persuasive value.

³⁵ See Essense Obhan, "Patenting of Software Inventions in India", *Spicy IP India*, available at:

<http://spicyipindia.blogspot.com/2009/02/guest-post-on-software-patenting-in.html> (last visited on 3rd August, 2009).

³⁶ See Shannahad Basheer, "A method to the madness", *The Mint*, available at: <http://www.livemint.com/2008/11/05233957/A-method-to-the-madness.html?h=B> (last visited on 3rd August, 2009).

³⁷ See Draft Manual of Patent Practice and Procedure, pp. 72 to 74, available at: http://ipindia.nic.in/ipr/patent/DraftPatent_Manual_2008.pdf (last seen on 4th August, 2009).

“Technical applicability of the software claimed as a process or method claim, is required to be defined in relation with the particular hardware components. Thus, the ‘software per se’ is differentiated from the software having its technical application in the industry. A claim directed to a technical process which process is carried out under the control of a program (whether by means of hardware or software), cannot be regarded as relating to a computer program as such. For example, “a method for processing seismic data, comprising the steps of collecting the time varying seismic detector output signals for a plurality of seismic sensors placed in a cable.” Here the signals are collected from a definite recited structure and hence allowable.”³⁸

In relation to the second category of apparatus claims, as long as it discloses a process limitation, that is it is of specific application, it would be patentable. It is interesting to note here that the manual states that “[a]s a general rule, a novel solution to a problem relating to the internal operations of a computer, although comprising a program or subroutine, will necessarily involve technological features of the computer hardware or the manner in which it operates and hence may be patentable.” This seems completely contrary to the legislative intention. The words *per se* must refer to the computer program in conjunction with something, as opposed to a mere computer program. A computer program having *technical* application is still a computer program *per se*. This sort of an interpretation could lead to absurd consequences. A computer program used for a non-technical process (having a non-technical application), but a process nonetheless, causing transformation in matter (for instance, as in *Diehr*, a process for curing rubber), would not be patentable. There is also no jurisprudential backing for such an interpretation. There is no basis for the patentability of a computer program with technical application as opposed to patentability of computer programs *per se*. Through this manual, the patent office seems to be attempting to circumvent clear legislative intention.

However, the definition of computer program has been taken to include all forms of software without any distinction whatsoever. The manual states –

“Computer programs are a set of instructions for controlling a sequence of operations of a data processing system. It closely resembles a mathematical method. It may be expressed in various forms e.g., a series of verbal statements, a flowchart, an algorithm, or other coded form and maybe presented in a form suitable for direct entry into a particular computer, or may require transcription into a different format (computer language). It may merely be written on paper or recorded on some machine readable medium such as magnetic tape or disc or optically scanned record, or it maybe permanently recorded in a control store forming part of a computer.”³⁹

This is perhaps one of the merits of the manual. The manual is also clear that algorithms (and mathematical methods) and computer programs that are simply expressed on a computer readable storage medium are not patentable.

Perhaps if one studies the words of the Section itself, some light can be shone on the legislative intent. A study of the provision will be undertaken in the next section as this paper seeks to compare the Indian and US position on software patents.

4. Comparison between Indian and US position

The Act puts a specific bar on the patentability of a mathematical method and of algorithms. It is clear therefore that a computer program as used in the Act has a distinct meaning from an algorithm or mathematical method. This seems to imply that any computer program *per se* would not be patentable as under the Act. As far as the mathematical method or algorithms go, this seems to be consistent with *Benson*. As held in *Benson* and *Flook*, a patent that pre-empts the use of an algorithm is not permitted. Clearly, this is covered in the Indian scenario. The more tenuous part relates to the interpretation of *per se* as it appears in the Section. If one were to apply the interpretation as in the manual, notwithstanding that it may be in contravention with the express words of the Section, the *technical* application test seems to be akin to the position of law after *State Street* and *A.T. & T.* All software that has technical application would certainly have a useful, concrete and tangible result. However, since the *technical* application test seems to be completely against the statute, the correct interpretation would lead us to a situation that is akin almost entirely, to *Bilski*. While software *per se* may not be patentable, software that is tied to a machine or that causes a physical transformation would not be software *per se* and hence, would not be hit by the bar under Section 3(k). The position in *Bilski* is the correct test that needs to be applied even in the Indian scenario. However, such an interpretation stifles the patentability of software. Perhaps this explains why the patent manual has sought to step away from this approach. Much like the CCPA and CAFC did in respect to the SCOTUS judgements that tied them down, similarly the manual could as well prove to be an attempt to shift the line of thinking to be something that is more pro software patent.

³⁸ *Ibid* at pp. 73.

³⁹ *Ibid* at pp. 72-73.

5. Conclusion

Bilski will soon be heard by SCOTUS and the law in relation to software as well as business method patents will hopefully be settled. The problem however, as seen in the past is that the CAFC will endeavour to distinguish or otherwise circumvent any anti patent ruling of SCOTUS. Given however, that *Bilski* seems to be walking the middle path, it might be in the interest SCOTUS to uphold it. Considering SCOTUS lately seems to be harbouring certain anti-patent sentiments,⁴⁰ it may not be surprising if they hold that *Bilski* has been decided correctly in that the only way for software to be patent eligible is to satisfy the test, that is, it is both a necessary and a sufficient condition. Perhaps what SOCTUS can do is to unequivocally extend the same to pseudo process claims as well. An extreme interpretation would be that the tests would apply across the board to all such forms of patent applications, regardless of form. For instance, the CAFC in the *A.T. & T.* case stated that the scope of § 101 would be “the same regardless of the form--machine or process--in which a particular claim is drafted”.

Perhaps the main problem with the law in the United States in relation to software patents is that the law is very old (drafted prior to the growth of the software industry and not amended since) and hence, the words are subject to varied interpretations given the new set of circumstances. However, in India, it has been sought by legislative mandate to bar all business method patents and bar patents that relate to software *per se*. The words are clear and unambiguous. India’s software industry is still in its initial phase and therefore, there is perhaps a need for allowing the patentability of software. The debate for or against patent protection is beyond the scope of this paper. However, the grant of patent protection for software that is already mandated copyright protection under the TRIPs agreement⁴¹ could prove to be extremely problematic. Software is perhaps more apt to receive patent protection than to receive copyright protection. For a change to be achieved in the mandate under TRIPs, countries should show in practice that patent protection has been useful in protecting the rights of the creators of the software, thereby doing away with the need of copyright protection for the same. This cannot happen in India unless the Indian Parliament amends the law. The patent manual if challenged in Court as being *ultra vires* the Act is likely to be struck down. Under law, therefore, there seems almost certainly to be a proscription on software patents. It is now up to Parliament to look at the policy implications of this and make changes if necessary.

References

- 1) Armon, O and E. Gardner (2009) “New Restrictions on the Patentability of Process Claims: Looking Beyond In Re *Bilski*”, *Journal of Internet Law*.
- 2) Carvalho, N.P. (2002) *The TRIPs Regime of Patent Rights*. London: Kluwer Law International.
- 3) Edwards, M.J. and D. Steinberg (2009) “The Implications of *Bilski*: Patentable Subject Matter in the United States”, 49 IDEA 411 (2009).
- 4) Fusco, Stefania (2009) “Is in Re *Bilski* a Déjà vu?” *Stanford Technology Law Review* (2009).
- 5) He, L. (2008) “In Re *Bilski* En Banc Hearing on patentable subject matter: Farewell to Business Method Patents?” *Boston University Journal of Science and Technology Law*, 252 (2008).
- 6) Hulse, R.A. (2009) and R.R. Sachs, “Making Sense of the Revived Machine-or-Transformation Test in Re *Bilski*”, *Intellectual Property & Technology Law Journal* (2009).
- 7) McEniery, B.J. (2009) “The Federal Circuit in *Bilski*: The Machine or Transformation Test”, 91 J. Pat. & Trademark Off. Soc’y 253
- 8) Obhan, Essense “Patenting of Software Inventions in India”, *Spicy IP India*.
- 9) Shimokaji, M.A. and P.L. Gahagan (2009) “Mind Over Matter: The *Bilski* Decision, Like Others before It, Reveals How Courts Have Frequently Kept Patent Law Lagging behind Technology”, *Los Angeles Lawyer*, 36 (2009).
- 10) Stobbs, G.A. (2000) *Software Patents*. New York: Aspen Law & Business, 2nd Edition

⁴⁰ See L. He, “In Re *Bilski* En Banc Hearing on patentable subject matter: Farewell to Business Method Patents?”, *Boston University Journal of Science and Technology Law*, 252 (2008).

⁴¹ See Article 10.1 of the TRIPs Agreement read with Article 9.2. In his book, the TRIPs regime of patent rights, Carvalho argues that patent protection is more apt for software as opposed to copyright. However, this comparison is beyond the scope of this paper. See N.P. Carvalho, *The TRIPs Regime of Patent Rights*, 148-149, (London: Kluwer Law International, 2002).