

SOFTWARE PATENTS IN THE INDIAN FRAMEWORK: AN ECONOMIC ANALYSIS OF PROBLEMS AND PROSPECTS

Meera Jayakumar*, A. Harsha Vardhan**

This article examines the nuances of the debate concerning the intellectual property regime governing software. It analyzes both legally and economically the question of whether software should be granted patent or copyright protection and with an eye trained on the Indian software industry, it argues for a stronger copyright regime by virtue of the considerations of time-period, extent and threshold of protection, and enforcement mechanisms.

I	WHAT ARE SOFTWARE PATENTS?	220
II.	THE ROOTS OF THE DEBATE	221
III.	THE LEGAL SQUABBLE	222
IV.	THE ECONOMICS	225
V	ANALYSIS AND CONCLUSIONS	227

I. WHAT ARE SOFTWARE PATENTS?

The term “software patent” does not have a definition as yet and all that can be done to define it will be based on individual surmise. This is one of the problems with the issue of software patenting, namely that the term lacks a universally sanctioned definition. Computer programs are basically divided into “application programs”¹ and “operating system programs”.² According to Richard Stallman, the co-developer of the GNU-Linux operating system and proponent of Free Software, “(s)oftware patents are patents which cover software ideas, ideas which you would use in developing software.”

This essay will elaborate upon the arguments of both sides in the long-standing debate of whether software should be granted copyright protection or

* III year, B.A. LL B (Hons.). Student, Gujarat National Law University

** III year, B.A. LL.B.(Hons.). Student, Gujarat National Law University.

1 Application programs are designed to do specific tasks to be executed through the computer.

2 Operating system programs are used to manage the internal functions of the computer to facilitate use of the application program.

patent protection. It will then conclude that copyright is better for protecting software from the *legal* and *economic* angle. This debate has assumed enormous significance in the Indian context, following the 2005 Amendment to the Patent Act, 1970 (which does not prohibit software patent expressly) and its possible effects on the economic interests of Indian software firms.

II. THE ROOTS OF THE DEBATE

Software has been remarkably difficult to classify within a specific category of intellectual property protection. This is because the characteristics of software are unique among protected intellectual creations, presenting particular difficulties for those drawing analogies with existing legal subjects.³ The problem lies in the fact that software is not a monolithic work or a "building block" invention. Each type of software consists of several elements that could fall within different categories of intellectual property protection,⁴ i.e. software is *complicated*. In addition, one piece of software may contain anywhere between 100,000 to 10 million lines of code, any of which might be patented. The complexity of software makes it dependent on many technologies.

Some authors argue that the very nature of the software industry is such that the concept of patents cannot be imported into it.⁵ The pace of development in software is very rapid. This means that the software industry is awash with ideas, and that innovators are outpacing each other everyday to create newer software. By the time an immature technology develops to the point where it can be incorporated into products, it has a dozen or more patents on it which render it commercially intractable. This creates an uncertainty in the development of new software.

This in turn presents a difficulty, as once a particular feature in a new software program is discovered to be infringing an old patent, the innovator has to work around the patent or even remove that feature entirely. In addition, the software industry is market-driven rather than being innovation-driven. The philosophy of the industry is "doing it right the first time" rather than just doing it first. Consequently, the rationale behind patents, which is to encourage invention for the sake of invention, is against the philosophy of the software industry and not in consonance with it. But

3. A. G. González, *The Software Patent Debate*, 1(3) J. INTELL. PROP. L. P., 196-206 (2006).

4. D.W. Carstens, *Legal Protection of Computer Software Patents, Copyrights, and Trade Secrets*, 20 J. CONTEMP. L. 13 (1994).

5. G. Irlam & R. Williams, *Software Patents. An Industry At Risk*, <http://osnome.che.wisc.edu/~epperly/epperly.html>.

the one snag which deserves intricate analysis is that the economy of the software industry is unlike any other industry which may benefit from patents.

A study of the Indian software industry shows that it is represented mainly by private domestic firms which, though large in number, are small both in terms of their assets and level of earnings. Only a handful of these firms have an asset size larger than Rs. 300 million. Similarly, very few earn revenues in excess of Rs. 500 million. This shows the significance of the need to have a regime of software protection that takes care of the needs of these small and medium sized enterprises (SMEs) which face great threats from the larger non-Indian players.⁶ Evidently, the choice is between the high level of protection that patents provide and the lower threshold of copyright protection. This choice cannot be made indiscriminately but must depend to a very large extent on the legal and economic aspects involved in granting either of these two kinds of intellectual property protection. This paper examines these impediments in some detail and the correlation between them before concluding that, in the Indian scenario, copyright protection is much more suited to Indian software SMEs.

III. THE LEGAL SQUABBLE

The debate on software patents and software copyright has been argued out on various levels. The legal aspects are a major battle ground for either side. Some of these aspects have been highlighted hereunder.

The major proponents of copyright protection for software argue that firms do not invent for the sake of getting patent monopoly. Surveys conducted by various scholars reveal that the innovation incentive of patents is hardly of any value, especially in the software industry.⁷ The vast majority of software patents are obtained by firms outside the software industry and with little investment in the inputs required to develop software inventions. In fact, industries known for prodigious patenting in general accounted for the vast majority of software patents obtained. This trend is also reflected in the patent propensity⁸ of firms in the software industry compared to those in other sectors.

6. C. Chakraborty et al., *Indian Software Industry: Structure, Trends and Constraints*, 1(2) J. SERVICES. RES. 73 (2002).

7. J. Bessen & R.M. Hunt, *The Software Patent Experiment*, (Federal Reserve Bank of Philadelphia, Working Paper, 2004), available at <http://www.rcsearchoninnovation.org/swpat.pdf>

8. Patent Propensity is the average number of patents obtained from a given amount of resources spent on developing new products and processes.

It was observed that firms apply for more patents when they are more capital and research intensive, and when their work force has more engineers and programmers.⁹ Consequently, manufacturers accounted for a whopping 75% of software patents in the 1990s compared to firms in the software industry which received 7% at most. This data seems to support the argument of the copyright-for-software school that patents do not provide a substantial incentive for inventing software. Hence, the very purpose behind granting patent is negated by this argument.

Another argument is that a patent covers more than the literal subject matter of the claim by including the prospective technology inherent in the earlier patent. This is the “prospective theory” of patent interpretation.¹⁰ Hence, what may be covered under the protection of the patent is not just the invention itself, but also anything that may be remotely related to the invention. The issue to consider is how just it is to exclude from the field of further research and development such incidental advantages due to the wide nature of patent protection. We can correlate this to the fact that the software industry mainly innovates on an incremental and cumulative basis and does not create building block inventions. Hence, the argument implies that granting a wide scope of protection to incremental invention negates the very philosophy behind patent protection, which is to encourage revolutionary innovation, instead of protecting incremental improvements in existing technology.

Economically, this argument can be brought under the “Prospective Theory of Patent Scope” wherein issued patents would operate as broad reservation of rights in the technical landscape. As a result, patentees could credibly seek to exact royalties for nearly all improvements, whether literally infringing or not, and improvers would need to think twice before refusing such demands. Second-comers would need permission to develop and market their innovations.¹¹

Another troubling aspect is the time period of protection granted to software if it is made patentable. The question of the appropriate time period for patent protection has, of course, been a point of contention in the public health and pharmaceutical arena, but now is the time to question the period of high quality

9. Bessen & Hunt, *supra* note 7.

10. S. Besen & L. Raskind, *An Introduction to the Law and Economics of Intellectual Property*, 1(5) J. ECON. PERSPECT 3-27 (1991).

11. E.W. Kitch, *The Nature and Function of the Patent System*, 20 J L & ECON. 265 (1977); see also J.E. Cohen & M. Lemley, *Patent Scope and Innovation in the Software Industry*, 89(1) CAL L. REV. 3 (2001).

and exclusive protection that will be granted to software if it is patentable. Under the current Indian patent regime, protection is granted for a period of 20 years. It is very evident from the nature of the industry and the pace at which it is developing that once some software is invented, it will last as a marketable product for a period of around two to three years before some other new, improved and modified version is invented by someone else.¹² The consideration in this case is the stagnation that will occur by virtue of the nature of the software industry and of the protection granted. No one will be entitled to use the patented software and create or invent around it, as it will amount to infringement. Hence, no one will be allowed to use that software and create a better or improved version.

An adverse consequence of the above mentioned argument will be on interoperability and standards of openness. The software industry thrives on these two characteristics of software. Interoperability and openness are essential for different programs to interact with one another.¹³ Once a software patent is granted, the holder will have the monopoly to produce all related software that facilitates interoperability. This will have adverse effects on the industry as a whole.

Confusion with regard to the subject matter of software patents itself is a major obstacle. The Indian Copyright Act has given no definition to "software" though computer programs are considered literary works. Also, there has been no case law with respect to software patents in India, making it very difficult to assess the judicial opinion on this debate. The confusion with regard to subject matter is evident from the attitudes of the courts in the USA, which pioneered the grant of patent protection to software. Two conflicting decisions of the Court of Appeals for the Federal Circuit (CAFC) in the cases of *In Re Grams*¹⁴ and *In Re Iwahashi*,¹⁵ highlight the uncertainty in the situation, when the CAFC held two similar claims involving similar algorithms to be non-patentable in one case and patentable in the other. The question is how a country like India, which is in its nascent stages of Intellectual Property protection, will solve such problems that go to the very essence of the debate, when patent pioneers like the USA are unable to solve them. The rejection of a proposal to implement a uniform law on software patent by the EU

12. Irlam & Williams, *supra* note 5.

13. This feature is absent in the current copyright regime, which doesn't provide for the publishing of the source code of software but which is mandatory once the patent is granted to the inventor.

14. 12 U.S.P.Q. 2d. 1824 [1989].

15. 12 U.S.P.Q. 2d. 1908 [1989].

also signifies the sketchy and murky character of this area of Intellectual Property protection.

IV. THE ECONOMICS

Though the main purpose behind granting patents is to encourage innovation, the economic feasibility of granting patents, especially to software innovations, may become an important aspect in deciding whether software should, in fact, be granted patent protection. The idea that the Software Revolution that began towards the last decades of the 20th century is a result of the incentives provided by the patent regime is an unlikely one. It will be seen that the innovation in the software sector had begun much before software was made patentable subject matter. So, it is highly improbable that patent protection played a major role in the financial and technological evolution of the software industry.¹⁶ The Bessen and Hunt experiment also shows that there is no empirical evidence supporting the relation between the number of software patents and R&D investments, meaning that permitting the patenting of software is not a useful way to stimulate innovation.¹⁷

In this context, an analysis of the likely economic costs which will critically influence the scope of protection is a priority. These economic costs are bound to affect patents and incentives in India, since the greater the costs of inventing, the lesser will be the incentive to invent. Hence, they are important indicators of the advantages of copyright protection over patents. This is due to the structure of the Indian software industry which is made up mostly of SMEs that have no financial potential to compete with larger multinational firms and companies. Software patents may just make it more difficult for such SMEs to protect their creations as the expenses incurred in acquiring and defending such protection may not be affordable for them.

What is required at the policy formulation level is a cost-benefit analysis of making software patentable, focusing both on the costs and the attendant benefits that might be the consequence of innovative activity by SMEs in India specifically. The author outlines some of these considerations.

The costs that will be accrue in terms of the prohibition on subsequent innovation that will ensue due to the restrictive nature of software patents is a

16. R.M. Hunt, *You Can Patent That? Are Patents on Computer Programs and Business Methods Good for the New Economy*, *Bus. Rev.* 5 (2001).

17. Bessen & Hunt, *supra* note 7.

major predicament for policy framers. It is often impossible to tell in advance whether a subsequent researcher's use of a patented invention will lead to an improvement falling within the scope of the claims of the prior patent or to a substitute technology falling outside the patent claims.¹⁸ The processes through which patent applications are reviewed are necessarily backward-looking, since they focus on prior art, much of which is embodied in previously issued patents. New fields of inventive activity therefore pose significant challenges to patent review procedures.¹⁹ The costs of conducting a patent search to find out what feature of an invention might or might not infringe existing patents must be considered, especially in light of the fact that patent application databases are extensively used due to the increasing number of software patent claims being filed and granted.

It may be technologically impossible to develop an alternative that is compatible with the existing network and falls outside the scope of the patent.²⁰ Hence the costs of licensing fees that may have to be paid if any developer wishes to use an obvious, yet patented, feature have to be taken into account.

Usually owners of software patents can sell or license their patents for cash, or start a business and use the patent to exclude competition. Patent income is strongly correlated with the usefulness of the invention claimed in the patent. But in the case of Indian SMEs, this income will count only for a very small fraction of their overall income. These costs intensify in the case of Indian SMEs which, though are the hubs of real innovation, cannot withstand the competition from larger companies. Another area of study which cannot be ignored is the costs incurred in terms of inventing over or around "deadwood" patents that prevent innovation due to their structure in the form of ingenious compilations of existing obvious or non-patentable technology.²¹

-
18. The uncertainty arises in part because it is difficult to predict the course and outcome of research projects, and in part because it is difficult to determine the validity and scope of patent claims until these matters are resolved in litigation. See R.S. Eisenberg, *Patents and the Progress of Science: Exclusive Rights and Experimental Use*, 56 U. CHI. L. REV 1017 (1989)
 19. J.H.G. Stuart & D.C. Mowery, *Software Patents: Good News or Bad News?* (Georgia Inst. of Tech., Working Paper, 2004).
 20. *To Promote Innovation: The Proper Balance Of Competition And Patent Law And Policy 22* (2003), available at <http://www.ftc.gov/os/2003/10/innovationrpt.pdf>.
 21. This may also be termed as costs accruing out of the existence of "trivial" patents which prevent pioneering innovation because of having patented obvious software features.

This will include costs sustained in designing around an existing valid patent on an outdated technology.

V. ANALYSIS AND CONCLUSIONS

Based on the abovementioned arguments and considerations, the authors conclude that copyright protection is more advantageous to the Indian situation. One advantage is that the copyright accrues as soon as the literary or artistic work is created. As of now, software is expressly classified as a literary work under S. 2(o), Indian Copyright Act, 1957. Hence, there is no need to file an application (called a specification in the case of patents) for acquiring the invention protection, but the right *in rem* accrues automatically, and may be defeated in court on appropriate evidence. This saves expenditure in terms of drafting and filing a complex patent specification as well.

Since copyrights can be granted if the creation falls under defined categories and exhibits a modicum of creativity, it is more suited to software by virtue of the fact that software-related inventions are more cumulative than building-block in nature. This sufficiently ensures that the creator's right over the software is protected, also enabling other inventors to develop and modify the existing creations by individually coming up with newer models. This is subject to the tests of infringement laid down by the Act as well as the courts, depending on whether such "working around" will constitute infringement.

Since copyright protection for software is uniformly recognised worldwide and there is an international copyright protection mechanism for software in the form of the Berne Convention of 1971, the Universal Copyright Protection Convention of 1971, and the Software Directive (as opposed to the lack of an international patent protection mechanism for the same subject matter) it is more beneficial for Indian SMEs to opt for copyright protection of their creations as such protection becomes enforceable worldwide.

In addition, opting for a copyright regime for protecting software allows indigenous creators to adopt defences like *scénés a fairé* and fair use, so that inventors may be allowed to work around existing cumulative protected programs that have been granted protection, but otherwise serve no useful purpose. This calls for the expansion of these doctrines in the judicial arena as they may currently be unsuited for application to software cases specifically, so that copyright protection for software may be made stronger. At least in the US, copyrights are granted very powerful protection by the courts. Injunctions are liberally granted and monetary damages awarded. Also, infringing material is often impounded. If similar jurisprudence is imported into India by the courts, it will ensure that the creative expressions of indigenous firms are sufficiently protected and

compensated for in case of infringement, especially by market players who wield a lot of financial clout.²²

Also, the very claim that copyright offers only a minimum threshold of quality protection may actually work in favour of the indigenous market for software. Since acts which amount to infringement of copyright are clearly defined in the Act, there can be no infringement if individual inventors come up with similar ideas for software independently. This in turn may work as an incentive for firms to create and formulate improved versions of software currently available. Also, this may help in creating standards of interoperability as new software can be improvised upon to suit and interoperate among various operating systems.

In conclusion, it is clear that legal arguments are in favour of software copyrights as opposed to patents by virtue of the considerations of time-period, extent and threshold of protection, and enforcement mechanisms. The economic burden imposed on Indian software SMEs will be far greater than the advantages that might accrue if a software patent is granted to them. Hence what is required is not an entirely new regime of protection for software in the form of patents, but improvements in the existing copyright regime. The idea of having a stricter and more stringent protection regime can be borrowed as a matter of jurisprudence from the USA, as has been mentioned above. But the absence of a stringent copyright regime for software does not justify the introduction of a higher threshold of patent protection.

22. Besen & Raskind, *supra* note 10.