PATENTING OF COMPUTER SOFTWARE IN THE CONTEXT OF GLOBAL TRADE IN IT SOFTWARE SERVICES

By

Pragya Srivastava

Corporate Lawyer and alumnus of Symbiosis Law School Noida, constituent of Symbiosis International University, India

Email: pragya.newdelhi@gmail.com

&

Dr Debashis Dutta

Chief Scientist, Reliance Industries Limited and Former Director General, Center for Development of Advance Computing (CDAC), Ministry of Electronics and Information Technology (MeitY), Government of India

Email: debashis.dutta@ril.com

KEY WORDS: Computer programme, Semiconductor, Intellectual property rights, Electronics, Circuit design, Algorithm, Data flow structure, Integrated circuit, Innovation, IT infrastructure, Policy, source code, object code, Patent Act etc

---

ABSTRACT

In this paper, an effort has been made to examine the patentability of computer software and propose a new tool for protecting intellectual property existing in computer software. To justify the logical reasoning to protect the computer software, a comparison has been made with integrated circuits to understand the reason behind creation of a separate

intellectual property tool for protecting intellectual property existing in integrated circuits. Based on discussions, it is generally found that present patent tool is not completely capable of protecting intellectual property existing in computer software. At the end of the discussion, a suggestion has been made to protect the data flow structure like topography of integrated circuits.

INTRODUCTION

The 19th and 20th century as well as the evolving 21st century have been known for the rapid pace of globalization, which is manifest in the growing cross national flows of goods, services, investments and ideas. These combined with the growing ease of imitation, produced a strong and continuing demand for improving the international legal framework for the protection and enforcement of IPR. IPR have thus moved rapidly from being an esoteric subject confined to specialist circles to become a major policy issue in international economic relations and a term recognized by the general public the world over.

Post independence, India has witnessed remarkable changes in the development process. Particularly, in the last two decades, our nation has made great strides in the area of information and communications technology. Young and talented minds have set many innovative examples driving the attention of think tanks and policy makers to re-examine the conventional way of thinking.

The Indian software industry has shown a tremendous growth with a Compound Annual Growth Rate (CAGR) of around 10 to 12% over the previous five years[i]. This impressive growth has been reflected in both the exports of the software and services sector as well as the surge in the domestic market. India has led the first phase of growth of the IT phenomenon due to some inherent advantages that it offers: abundance of talent, superior delivery quality, cost advantage and favorable policy interventions by the government towards IT infrastructure along with other growth-oriented policy moves.

It is an established fact that economic growth exists in knowledge-based industries. There is significant revenue potential to be recognized through innovation leverage in software sector. The Indian software and services sector can tap additional revenue streams worth ~50 USD Billion through the innovation itself[ii]. With the success of the Internet as a fusion of computer science, electronics & communication and information systems, the requirement of physical presence is receding. This has enabled even small software companies to become competitive by exhausting their resources in the knowledge based products development instead of investing in the tangible infrastructure development. Since in computer industry the raw material and the final products both are made of knowledge, computer software is known as intangible of intangibles. This unique application of knowledge in the industry and the economic growth prospects attached with the products, intellectual property rights protection becomes a basic requirement. Ironically till date, not all facets of computer program are protected and the government of India, like most of the governments of the world, has imposed a particular tool for protecting intellectual property existing in a computer program.

Pamela Samuelson writes in her article[iii] that *Computer programs have a number of important characteristics that have been difficult for legal commentators and decision makers to perceive. First, the primary source of value in a program is its behavior, not its text. Second, program text and behavior are independent in the sense that a functionally indistinguishable imitation can be written by a programmer who has never seen the text of the original program. Third, programs are, in fact, machines (entities that bring about useful results, i.e., behavior) that have been constructed in the medium of text (source and object code).*

The most crucial issue for the computer software industry is that many of the products and services produced involve sizeable amount of investment and due to lack of proper protection often pose a financial risk to develop[iv]. Unlike other products they are easy to copy at the cost of bare investment. Ease of copying creates major disincentives to develop new and innovative software programs, hindering the development of the

innovative knowledge industry. The importance of computer software in the contemporary world needs secured investment in the computer software development industry. This is possible only by ensuring the protection of knowledge wealth behind the creation of programs/software, which is possible by using the existing IPR tool optimally and devising new tools as and when required.

The government of India has taken several initiatives, recent being issuing guidelines for the Examination of Computer-Related Inventions (CRI) with a view to have a well-organized examination process for patent applications relating to CRI[v].

On the 30th day of June, 2017, the Controller General for Patents, Design and Trademark[vi] issued revised Guidelines for the Examination of Computer-Related Inventions (GECRI) with an aim to clarify the exclusions provided under section 3(k) of the Patents Act so as to ensure more efficient examination for other patent applications relating to CRIs.

The GECRI does not constitute any rulemaking. Its main aim is to provide guidelines for the examination of patent applications in the field of CRIs by the Indian Patent Office so as to further foster uniformity and consistency in the examination of such applications. The document was drafted with the basic objective of bringing out clarity in terms of exclusions expected under Section 3(k) so that eligible applications of patents relating to CRIs can be examined speedily.[vii]

NEED FOR A SUITABLE SOLUTION:

Whether patents are sufficient?

The US statute, unlike Indian statute, is silent on Computer Software patenting. Computer software patents have found the way into the patent office records through courts. *The European Patent Office (EPO) rejects the application if in the claim(s)*

software is described as a product, but if the wording computer software related invention is used there are chances to expect positive response from the EPO[viii]. Since there is no coherence between the statutes and the case laws, computer software patenting process is often a long, frustrating and costly affair. The case laws exist only in a few developed countries; so outside their jurisdiction these case laws have only guiding source value. Even if computer software is accepted as a subject matter of patenting, the future of software patent remains at stake in countries like India. The patent laws of India ensures that computer software patent application fall in one or the other four categories of non-patentable items described in the sub-section 3(k) of the Patent Act and thus remain un-patentable.

It is an established fact that functional aspects of computer programs cannot be protected by copyright tool[ix]. There are certain drawbacks in the patent tool, which is the reason why patent tool is not accepted as the appropriate tool for protecting the computer programs. The following reasons are often cited for rejecting patent tool for protecting computer programs:

1. In comparison to the life of a computer program, the process of patent grant is very long. Generally a computer program becomes obsolete in three years time and in most of the countries patenting procedure takes at least three years time. So by the time, if in case, patent is granted in three years time the software completes its life.
2. Software protection under the patent system is inappropriate because it is available only after the software undergoes a costly and time-consuming examination[x].
3. Most of the countries do not define the meaning of *per se* and therefore do not differentiate between compute program and computer program *per se*. This approach forcefully puts computer programs into the domain of copyrightable subject matter.
4. The patent rights subsist in the claim part of the patent specification and it required expert opinion to understand the domain of the rights existing in the

specification. If proper care is taken of in not infringing someone else's patent rights, it would be very obscure task with the development of a compute program.

5. The success of the 'open source' business model is posing a big challenge to the proprietary software[xi]. Who shall be the infringer of patent rights existing in the proprietary software if violated by an unknown person who downloads and modifies an open source program?
6. Software developed on open source software business model does not require patents as they are based upon different philosophy. The IPR regime is based on the concept of personal property and therefore ownership vests in some person, which is not true in open source software model.

Exploring a viable Solution

For protecting software, copyright and patent both of the tools are not sufficient, even if applied together; then, what is the way for protecting computer software? A new tool may satisfy the urge of protecting 'proprietary' software effectively.

One of the interesting and logical approaches for finding a method for protecting IP existing in computer software would be analyzing IP protection methodology taking the case of Integrated Circuit (IC) topography[xii]. Commercialization of IC chips started in the decade of 1970's. The IC research and development cost was very high and the technology was confined to a few developed countries. *The development of a new family of new computer chips can cost over £ 10 million. A chip can be copied, however, for less than £ 100,000*[xiii]. Economic interests forced the investors to take initiative in securing knowledge placed in IC chips. The imitation risk was very high. It was possible to produce similar IC at one-tenth cost of the original IC production cost by the process of reverse engineering.

Treaty on Intellectual Property in Respect of Integrated Circuits for protecting layout design came in the year 1989 and it managed inroads into the TRIPS Agreement[xiv]. The TRIPS Agreement's binding nature forced the WTO member countries to make

provisions for protecting IC layout design (topography) for respective domestic IP laws of the Member countries.

The treaty on IP in respect of IC or the TRIPS does not suggest specifically for a separate tool for protecting topography of IC; it rather says that any IP tool like, copyright or industrial design, may be applied for topography. India, like many other countries, came out with the Semiconductor Integrated Circuit Lay-out Design Act, 2000, a new tool exclusively for providing protection to IC topography[xv].

Analogy between Integrated Circuits and Computer Software

An IC is considered as a logical equivalent to computer software. *Any operation performed by software can also be built directly into the hardware, and any instruction executed by the hardware can also be simulated in software.*[xvi] This analogy is true only theoretically. Practically, many considerations, like cost, modifications and compatibility with other software/hardware, play a role in deciding to choose the appropriate hardware or software means for achieving an objective. The similarity in nature provides an opportunity to compare software with IC in terms of IP protection.

Just like a computer program, an IC is a step-wise-procedure for implementation of one or another algorithm. Like computers, electronic switches/logic gates understand nothing but zeros and ones. Mathematics is at the root of the development of electronics and computer science streams[xvii]. Any task assigned for the execution by either an IC or a computer program needs instructions in zeros and ones (machine language). Though the programs are mostly written in fifth generation language (5GL) programs need to be converted into machine language by a compiler for execution by the machine. Hence, mathematics is the basic chemistry of electronics and computer science and merely due to this reason, innovations in these fields cannot be stopped from qualifying for patents. Though natural laws are beyond the

reach of private monopoly, all the inventions are practical applications of one or another natural law.

The basic character of both IC and computer software is the same. Both perform their desired task by the means of logic gates. The only difference is that IC performs as hardware and computer software performs in simulated medium. An algorithm can be written in different ways for performing the same objective. Therefore, it is possible to produce different IC chips and software for achieving the objective. But by protecting topography, no one can copy the same data flow.

The decision to put certain functions in hardware and the rest in the software depends upon cost, speed and amount of memory required, frequency for updating and reliability[xviii]. Otherwise there is no difference in designing the sequence of tasks by hardware but denying the same protection to software.

DESIGNING A NEW TOOL

Data flow systems where the schematic diagrams are portrayed which depict as to how the raw data travels from the entry point of an IT system/sub-system to the finished/usable data at the point where a user can interpret the data in the practical world and can make a definitive judgment about the predictability of the result. The data flow system may compose of a number of sub-systems and foundation layers.

Protecting Data Flow Structure

The development of a perfect data flow structure is the most crucial part of development process for software[xix]. By securing the data flow structure, all the ways of infringing the rights existing in the data structure can be blocked. Hence even by changing the language of the software, it would not be possible for using the protected data structure without infringing the IP rights.

Since it has already been accepted by devising a separate tool for protecting IC topography, there is no need to prove that data structure needs a separate tool for protecting IP existing in computer software. The IC tool can be modified for protecting software data flow structure. The data flow structure protection offers a holistic protection for computer software and is an answer to the questions mooted for and against patentability of computer software[xx].

Just by reading the instructions written in a program a programmer can draw program's data flow structure. This means that without applying extra efforts and legal professional's help understanding the software claims, a programmer can himself know whether a program is based up on the data flow of another program. Secondly, the data flow structure assures those who think that by the means of computer program algorithms, mathematical and business methods can be monopolized. Data flow structure assures that only application of an abstract concept is going in private monopoly. Since only application developed for the program is the domain of private monopoly created by exercising the rights, the abstract part remains open for the others for free use[xxi]. Thirdly, the data flow structure protection does not require complexities for claims writing and other patentability conditions and formalities. It is easy to be applied and an easy to be enforced method for protecting IP rights existing in computer software.

For protecting the data flow structure the strategy adopted for protecting the IC topography can be applied. This tool can be clubbed with the IC layout protection. The only difference between the two tools is that software are perfect intangibles and the IC layout designing is in tangible form, otherwise both are logically same and theoretically equally good for performing any task.

IN CONCLUSION:

The US and Japanese courts have explicitly cleared the software for patenting and in many other WTO Member countries software patenting is done by using computer-related invention terminology. If an algorithm is a part of a computer program, it should be included as a means to achieve some sort of objective to execute the desired goal stated in the program codes. Here the functional aspect of algorithm, not the algorithm itself, becomes a part of a patentable computer program. It is a matter of fact that computers understand nothing but 0 and 1; hence for the execution of each and every command by the computers the credit goes to the applied algorithm. Since algorithm is not patentable, it remains liberated from private monopoly bonds. What will happen when computers would start understanding other numbers also? Will that learning of computers would also be not a revolutionary invention? Will that also be opposed for patenting? Well, only time has answer to this question!

The intellectual class of the world does not accept computer software patenting because human intellect create them. But the same intellectuals accept patenting of gene based inventions. The simple reason is the fact that we are yet to discover the behavior of gene encoding and to uncover the triggering of bio-chemical reactions based upon the gene encoding.

Referring various case laws, it is interesting to note that no one tested software on patentability tests (novelty, non-obviousness and utility) in the courts on the issue of computer software as patentable items. It indicates that baring the monopoly over the abstract matter objection, there was no other objection in the computer software patenting.

Subsection 3(k) of the Indian Patent Act is the subject matter of this work which deals with what would not be considered as patentable subject matter. It lists mathematical methods, business methods, algorithms and computer programs *per se* as disjunctive non-patentable items. Courts have settled that business methods exception is not required in the patent law. It has been already discussed that business methods satisfying the patenting conditions are patentable. It is worth noting that in this work

business methods have been construed in relation to the computer software. The mathematical methods and algorithms are also not required in this subsection because in the abstract format they cannot pass the patentability tests and in the application format, if they satisfy the patentability tests, there is no objection in accepting them as patentable items. In relation to the computer program *per se*, it has been clarified that computer program *per se* means listing of the program and listing is expression of sequential commands understood and obeyed by the computers. They are clearly copyright subject matters. Therefore mentioning of computer program *per se* is not required in the Patent Act, as it only creates confusion as if computer programs are not patentable; but with the other three exceptions it is not possible to qualify computer programs as patentable subject matter. Hence, in the good faith of the development of the computer industry, it is prudent to amend the Patent Act accordingly. This will also satisfy the requirement of the Article 27 of the TRIPS Agreement.

The proposed tool is easy to include in the existing IPR regime. The proposed tool invites no troubles, like monopoly over the abstract matter. It can give an added advantage to the patented program and if patent is rejected or not allowed in the IPR regime it can provide protection independently. The easy availability, clear application and crystal clear domain makes data flow structure protection a worthy tool for protecting intellectual property existing in computer software.

Computers need software to run and development of computer software need investment and investment requires security of returns and security of returns need fool proof laws. This complete chain of sequence requires provisions in the IPR regime for protecting the intellectual wealth existing in the computer software.

REFERENCES

[i] Source : Nasscom (www.nasscom.in)
[ii] Discussions with NASSCOM research and innovation team
[iii] Pamela Samuelson *et all, a manifest to concerning the legal protection of computer programs*, Columbia Law Review

[IV] https://www.aicpa.org/InterestAreas/FRC/AssuranceAdvisoryServices/DownloadableDocuments/ASEC_Whitepapers/Risk_Technology.pdf

[V] http://www.ipindia.nic.in/writereaddata/Portal/IPOGuidelinesManuals/1_86_1_Revised__Guidelines_for_Examination_of_Computer-related_Inventions_CRI__.pdf

[VI] http://www.ipindia.nic.in/

[vii] Guidelines for Examination of Computer Related Inventions (CRIs), p-3, para-1.3.

[VIII] http://www.epo.org/

[IX] https://www.wipo.int/copyright/en/activities/software.html

[x] According to Leo J. Raskind of the University of Minnesota School of Law.

[XI] https://opensource.com/business/13/12/using-open-source-software

[XII] https://www.wipo.int/patents/en/topics/integrated_circuits.html

[xiii] Chris Reed and John Angle (Editors), Computer Law, p. 166, 4th Edn., Universal Law Publishing House, New Delhi, First Indian Reprint 2002.

[XIV] https://en.wikipedia.org/wiki/TRIPS_Agreement

[XV] http://www.mondaq.com/india/x/28601/technology/Semiconductor+Integrated+Circuits+Layout+Design+In+Indian+IP+Regime

[xvi] News and Comments, *LEGISLATION: HOUSE OF PANEL HOLEDS OVERSIGHT HEARING ON COMPUTERS AND INTELLECTUAL PROPERTY*, BNA's Patent, Trademark & Copyright Journal, Issue No. 956, November 16, 1989.

[XVII] https://www.cise.ufl.edu/~mssz/CompOrg/CDAintro.html

[XVIII] https://www.nap.edu/read/4761/chapter/12

[XIX] https://www.umsl.edu/~sauterv/analysis/Fall2010Papers/varuni/

[XX] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3972398/

[XXI] https://academic.oup.com/jla/article/9/1/51/3572441