Open Source Software Paradigm and Intellectual Property Rights

Vikrant Narayan Vasudeva†
Indian Law Institute, Bhagwan Dass Road, New Delhi 110 001, India

Received 27 March 2012, revised 16 May 2012

Open source software represents a paradigm shift in the field of software development. This new community based software development model, instead of relying on the conventional proprietary model of limited access, invites programmers globally, to freely copy, share, and modify the software. It is a misconception to believe that the general approach of open source software (OSS) towards IP laws and focuses on specific issues that emanate from the interaction of the OSS model with the existing intellectual property rights structure. How IP open source programs are placed in the public domain; they are very much protected by intellectual property laws, but distributed under terms which instead of being restrictive promote access. This way it challenges the established norms of all existing branches of intellectual property. The open source movement necessitates scrutiny; more than just being a new fangled approach, it catalyses debate regarding both the mode of software production and its protection. It is being used to propel arguments to revisit intellectual property jurisprudence. After all, the intention of the intellectual property-software system is to catalyse innovation and ultimately serve the society. How IP impacts on the OSS model and how OSS uses IP in a novel way to achieve its ends are discussed in this article.

Keywords: Open source software, intellectual property rights, license, proprietary, copyleft, open patent

The institution of open source has its roots in an ethical rebellion of sorts by software programmers expressed in the form of sharing source code of computer programs. The cause was subsequently taken up by academics and practitioners alike, and eventually this model snow balled into a parallel regime in the software development landscape.

Open source software represents a paradigm shift in the field of software development. As opposed to the generically referred ‘conventional,’ ‘proprietary,’ ‘closed,’ source code model, the open source software model emphasizes on unrestricted accessibility to the source code of the program. Also, unlike, proprietary software, open source code development is not a solitary or a closed group task – it is a community based development model adopting the ‘bazaar style,’ as opposed to the ‘cathedral’ approach. The open source development model is not subject to prejudices based on maturity, education or experience. Also, there is no need to assemble the community at one physical place, virtual interaction is sufficient. Thus, the open source model allows larger groups to interact, and increases accessibility to many more resources, while keeping the transaction costs at a minimal.

The open source movement traces its origin to two academic institutions—the Massachusetts Institute of Technology and the University of California, Berkeley. The open source philosophies that evolved at these two institutions, eventually formulated the polar factions of the movement represented famously by the GNU General Public License (GPL) and the Berkeley Software Distribution (BSD) License. Almost two decades later, Netscape Corporation was responsible for creating the third pillar bolstering the open source movement represented by the Mozilla Public License (MPL). Thus, open source software does not simply mean freedom of access to source code. It includes much more depending upon the standard setting body spearheading the project.

Though acronyms for the entire field of such software have been developed, viz. ‘Free/Libre/Open Source Software (FLOSS),’ ‘Free/Open Source Software (FOSS),’ ‘OS/FS’ etc., with time, simply the generic term Open Source Software (OSS) has come to encompass the entire field of software based on this particular mode of software development, whether it officially falls under the criteria of ‘Open Source’ software or ‘Free Software’ or ‘Libre Software’ or any other. Need for correct nomenclature distinction only arises when the licenses are being critically

† Email: vikrantvasudeva@gmail.com
analysed. In this article too, the term open source software has been used in its generic-industry recognized and popularized form as representative of the entire development model.

It is a misconception that open source programs are placed in the public domain; they are very much protected by intellectual property laws, but distributed under terms which instead of being restrictive promote access. This new model of software development, instead of relying on the conventional proprietary model of limited access, invites programmers globally, to freely copy, share, and modify the software. This way it challenges the established norms of all existing branches of intellectual property (IP).

The open source movement necessitates scrutiny; more than just being a new fangled approach, it catalyses debate regarding both the mode of software production and its protection. It is being used to propel arguments to revisit IP jurisprudence. After all, the intention of the intellectual property-software system is to catalyse innovation and ultimately serve the society.

This article introduces the general approach of OSS towards IP laws. It focuses on specific issues that emanate from the interaction of the OSS model with the existing intellectual property rights (IPR) structure. How IP impacts on the OSS model and how OSS uses IP in a novel way to achieve its ends are discussed. Thereafter, this article introduces how OSS propagates its model through the licensing medium. The essence of development of the OSS model is based on its marked variance from the proprietary licensing model. Finally, the article highlights the scope for complementary protection measures and licensing too are applied. It is not necessary that all the forms of protection would be exercised with every software; developers may choose to rely on none, some, or all of them.

Interaction of the traditional intellectual property structure and computer software generated the proprietary model of software development, which was the veritable apotheosis for over two decades. Laments increased that traditional IP protection in software were steadily whittling down the public domain. The key critique of the traditional manner of application of IPR to software is that it is better suited for protecting, than for facilitating relinquishment of rights. Open source philosophy when applied to software re-imagines the interaction and interpretation of various IP laws.

Trade Secret Protection

Trade secret law was favoured in the early phases of computing technology, when software was individually distributed under tight contractual control. However, with technological evolution, it no longer remained apt or adequate. Though trade secret protection is still used for software, it is no longer its dominant or sole mode of protection. Open source software furthers this rift; its ideology is in complete conflict with trade secret law, thus rendering its application to open source software unfeasible.

The Free Software Foundation (FSF) considers distribution of code as trade secret a violation of GPL. In fact, the FSF considers distribution under any type of restrictive agreement like a non-disclosure agreement a violation of the GPL. Thus, it appears that open source software and trade secrets are in perpetual conflict. However, the GPL allows developing code on a restrictive basis in a client-developer type software development model. Thus this is one scenario where trade secrets might still be applicable in the open source software context. Still, in such a scenario, the developer would be able to harness the open source community only till the period before taking the code as trade secret and any inputs thereafter would be curtailed. Hence, there are extremely limited possibilities for trade secrets and open source software to co-exist.

Trade secrets in the open source arena have been raised in litigation too in Red Hat v SCO, where, Red
Hat sought to obtain a declaratory judgment arguing that Linux being publicly available could not be protected by trade secrets, and hence, Red Hat had not violated SCO’s trade secrets.\(^9\)

**Copyright Protection**

Open source proponents disapprove of the current ‘proprietary’ model of copyright protection for software. They perceive it as a deviant application of the justification for copyright law - to induce innovation; also they challenge the need for monetary incentives itself to induce innovation in software creation. Open source proponents believe that the proprietary software model restricts the benefits to society.\(^{10,11}\) In an effort to rectify this perceived erroneous state, the open source regime introduced its own vision of copyright protection for software.

OSS innovatively re-interprets the essential legal foundation upon which the proprietary software industry exists. Unlike proprietary software, in which copyright is used to ‘exclude’, open source offers a dissenting logic to promote ‘inclusion’.\(^{12,13}\) The open source modus operandi essentially is to copyright software and then mass license it for use, improvement or modification, primarily under two models; one where the alterations are contributed back to the software community on reciprocal terms (colloquially referred to as copyleft) and second, where they need not be. This development model allows subsequent users to disseminate the open-source tenets which proliferate exponentially with increased distribution.\(^{14}\)

Herein lies the innovativeness of the open source premise, where some traditional copyright benefits are relinquished, thus allowing greater leeway in rights usage.

Scope of copyright protection has an inverse relationship with public access. Consequently, a need arises to determine the scope of copyright protection for computer software. However, it is easier said than done. Unfortunately, due to their different ideologies, and the manner in which open source software uses copyright law, any change in the law that strengthens open source software also promotes abuses in the proprietary system, and any change that directly addresses those abuses tends to weaken it.\(^{15}\)

Specifically, as regards the idea-expression dichotomy, ‘[n]obody has ever been able to fix that boundary, and nobody ever can.’\(^{16}\) Consequently, determining non-literal infringement as against literal infringement is difficult. Judiciary is divided as regards the scope of protection to be afforded to the non-literal elements of a computer program. The broader view provides expansive protection, bringing such elements as the program’s structure, sequence, and organization within its scope.\(^{17}\) On the other hand, the narrower view, confines the scope of protection by excising the non-copyrightable elements before determining the computer program’s overall copyright protection.\(^{18}\) Both factions have their share of critics. Whether the broader view or the narrower view is resorted to, both have their iota of positive factors and shortcomings, and both would cause the development of open source software along different planes.\(^{19}\)

A major concern for the open source arena and a common allegation of the proprietary arena is the infiltration of open source code in proprietary code and vice-versa. It could open both sides to allegations of infringement. This issue came to a heated point in the *SCO v IBM* litigation,\(^{20}\) where SCO claimed that IBM illegally incorporated SCO’s proprietary UNIX code into the open-source Linux operating system, and thus every Linux distributor, developer, and user would become copyright infringers if they did not pay a licensing fee.\(^{21,22}\) The court ultimately determined that 326 lines of code in the Linux kernel were potentially infringing.\(^{23,24}\)

Worse still, for the proprietary arena, if the viral taint of the open source license is applicable, it could well mean disclosing the code to the community at large. A possibility exists that incorporating code licensed under a reciprocal license like GPL into the source code of proprietary software could potentially ‘infect’ the proprietary software, thus causing it to be automatically licensed under the reciprocal license’s terms.\(^{25}\) This is what is speculated to have occurred in the case of Microsoft, where it was forced to contribute code (Hyper-V drivers) to Linux to avoid a GPL violation when it was discovered that Microsoft had incorporated GPL-licensed components into its Hyper-V drivers.\(^{26,27}\)

Open source movement does not relinquish copyright, infact, it is entirely dependent on copyright law.\(^{28}\) Without copyright protection and the ensuing licensing scheme, the OSS development model would be reduced to an honour system.\(^{29}\) However, the OSS movement reflects the intent of its founders to turn traditional notions of copyright, licensing, development and even ownership on their heads, even to the point of creating the term ‘copyleft’.\(^{30}\) In this regard, the GPL has been referred to as ‘a hack on the
copyright system' because it uses the exclusive rights under copyright law to promote rather than restrict access.\textsuperscript{15} This attempt to allow the author and the public to interact in a way that normal copyright does not is viewed as an inherently imperfect solution by critics to correct the unreasonableness of contemporary copyright law.\textsuperscript{31} Imperfect or not, this concept has spawned a completely divergent policy model of software creation. Moreover, the success of OSS licenses indicates that the existing copyright law is capable of facilitating cumulative innovation.\textsuperscript{32}

Moral Rights

Moral rights are considered extremely relevant as regards open source software.\textsuperscript{33} Largely, the open source software approach is based on attribution and reputation rewards as opposed to monetary rewards.

Moral rights in software are controlled either jurisdictionally, or by limits to the rights themselves.\textsuperscript{34} While some countries recognize moral rights, others do not. Some countries specifically deny moral rights protection to software authors, or permit authors to contractually waive moral rights; others curtail the moral rights available to software authors. Section 57 of the Indian Copyright Act recognizes the right of paternity and integrity in context of computer programs; however, the integrity right is curtailed by Section 52 in context of adaptations of computer programs for interoperability and back-up copy purposes.

Legal systems that recognize moral rights may already provide a degree of latent alternative protection for OSS; in such countries, violation of the open source licensing terms would constitute a violation of the developer’s moral rights.\textsuperscript{34} For example, under German law, despite the broad scope of release of OSS by the GPL, the modifier’s action can still be restricted by the author based on Section 14 of the Copyright Act in exceptional circumstances; by virtue of law, the author retains a right to prevent distortions or impairments of his work.\textsuperscript{35}

Open source licensing implements in a controlled manner the spirit of moral rights. Both GPLv2 and GPLv3 recognize the right of attribution subject to contractual provisions. Other licenses too carry similar provisions. The right of modification is not a waiver of this right.\textsuperscript{36} Thus, this showcases the inherent conflict between the freedom which a licensee of OSS strives for, to change that work any way he pleases, and the desire of the original author to obtain proper attribution.\textsuperscript{37} Analogously, it also allows a degree of control where an intentionally low quality modification could be held to infringe the reputation rights of its previous authors.\textsuperscript{38}

Patent Protection

Software patents are viewed with a jaundiced eye by the open source community.\textsuperscript{39-40} Open source proponents seek to revisit patent jurisprudence in the context of software programs altogether, citing it as an undeserved reward.\textsuperscript{41} and disputing the traditional incentive structure especially in view of open source software’s success.\textsuperscript{32}

Arguments against software patents range from the procedural to the theoretical. The procedural arguments vilify the patent office’s patent grants and procedure in computer software. They challenge the prolonged patent term, relaxed standards of non-obviousness for building-block programs, opaque prosecution process, non-disclosure of source code, abuse of continuation filings etc.\textsuperscript{40-41,43-44} The theoretical arguments, perceive software patents as the very antithesis of innovation by creating a ‘thicket’ – an ‘anticommons’ by which large number of ‘building block’ programs become legally inaccessible.\textsuperscript{41} While the proprietary model’s corporate and traditional intellectual property based structure creates avenues to address these concerns, the open source structure has limited options.

Furthermore, software patents pose a veritable threat to the open source propagation model. If an infringement of a proprietary program by an open source program, even a minimal bit is confirmed, it would essentially bring its development to an effective stop, at least in the open source arena. The open source faction’s woes are further augmented by the fact that software patents promote a form of veiled infringement of open source products on the one hand while conversely the very nature of open source programs make them susceptible to patent monitoring.\textsuperscript{41}

Largely, software patents are deployed as an indication of the company’s standing; they also provide enhanced collateral and negotiating power in the corporate arena. Threat of software patents does not dent large corporate houses’ indulgence in the open source arena; IBM, Phillips, Sony, Samsung and Nokia, to name a few, have express interests in open source software development, nonetheless, these same companies maintain their traditional corporate approach as well and continue to acquire software patents. However, software patents’ threat do have an intense moral and economic debilitating effect on
individuals and small developer companies, which are the real backbone of open source innovation and proliferation.\textsuperscript{45-46} A substantial amount of open source software has been affected by such patent threats and removed from public space.\textsuperscript{47-49} ‘In effect, large corporate stakeholders have erected a fence around pieces of open source territory.’\textsuperscript{50}

The open source community has made several effective forays to tilt the focus from building patent portfolios for corporate bulwarking, to emphasis on innovation in the technology sector. The open source movement is shoring its resources to combat any potential threat through some pretty innovative techniques viz. license provisions, creating patent pools, establishing prior art databases and emphasizing on proper prior art examination. Corporate goodwill from some proprietary manufacturers too has been extended to the open source community in this respect, which may have more to do with achieving versatility in business models.\textsuperscript{51-53}

Through license provisions, the open source software approach seeks to extend the effect of the copyleft reciprocal obligation from copyright to patents terming the concept as open patent movement.\textsuperscript{54} Thus, the software developer also grants permission to users to practice the patents held by him, when distribution of the software occurs. The work could be used as it is, or improved, in which case the patent improvement would have to be re-licensed to the institution that holds the original patent, and from which the original work was licensed. It also provides immunity to the licensees from infringement threats.

While the core concept of application of the copyleft reciprocal obligation to patent law is relatively straightforward, the practical difficulties are several. Firstly, there is a fundamental difference in the treatment of improvements under copyright law and patent law. This is primarily because of the copyright owner’s statutory entitlement to control improvements i.e. derivative works of a copyrighted work; a patent on the other hand does not afford any particular right to advances.\textsuperscript{35} Secondly, common law copyright protection is instantaneous as from creation, but patent protection requires registration. Obtaining patent protection on each improvement made would be extremely cumbersome and could effectively delay or bring the OSS project to a halt.\textsuperscript{67} Thirdly, unlike copyright, obtaining patents is a tedious and expensive process; hence, a lot of developers would be reluctant to freely release patents to the community.

Thus, though certain successes have certainly been achieved, in motivating the industry towards patent sharing,\textsuperscript{56} still, despite this initiative at correcting patent law in a similar manner as copyright law by the open source faction, it might be more difficult to achieve, and not gain a similar amount of popular support.

\textbf{Trademarks}

The term ‘open source’ was not trademarked prior to its widespread popularity as a descriptive term referring to itself.\textsuperscript{57} However, open source developers showed considerable sophistication in their use of trademark law, instead adopting certification marks to indicate if a particular software complies with the open source scheme, for e.g. the ‘OSI Certified’ mark affixed by the Open Source Initiative for software complying with their Open Source definition. Additionally, though not registered, the term ‘Copyleft’ in itself has acquired a distinctive trade use.

Using certification marks as opposed to trademarks gives recognition to the open source approach as opposed to a particular OSS product. This also avoids several hassles under trademark law like policing of the mark and ensuring proper attribution while allowing greater flexibility in usage and shifting equal onus of responsibility on all the developers.

\textbf{Existing Contractual-License Protection and Open Source Software}

The legal instrument for propagating the open source philosophy and its interpretation and application of IP laws is the license. The open source movement designed a counterintuitive licensing system based on the same legal premise as proprietary software but to different ends. Yielding IPR through the means of licenses, the open source faction promotes functional freedom for software, for developers and users alike. Ultimately, the OSS license is a specialized application of the conventional software license.\textsuperscript{58}

Though open source proponents are in agreement regarding the essential philosophy of the movement, when it comes to transcending policy into practice, licenses tend to diverge on certain terms important to collaboration. To avoid conflicts, and promote modulated growth, definitional standards have been set by prominent open source proponents like the
Open Source Initiative and the Free Software Foundation. Certain licenses like the Mozilla Public License, have set standards not by laying out definition parameters but by commercial practice. Licenses may have different terms, but they must each comply with the core principles propagated by these standard setting bodies in order for the software to be termed open source software.

This standard setting initiative is commendable, yet many complications remain. The licenses have different approaches. Many licenses have not been updated in years to reflect current trends. Interpretation issues especially with different jurisdictions exist leading to misapplication. Friction exists in license interaction leading to compliance issues. Existence of several standard setting organizations causes further impediments.

Moreover, some critics are of the view that all software licenses are vitally and intrinsically flawed as instruments of open source software development. Contract law is perceived as a ‘stopgap measure’ to patch broken copyright jurisprudence. The licenses are criticized for attempting to privatize intellectual property law, enhancing transaction costs, causing proliferation and confusion as regards interpretation and application. On the other hand, certain open source factions repudiate the existence of a contractual relationship altogether placing reliance entirely on copyright law.

It is undeniable that instead of a better license, a better approach is required. However, for the time being, as with proprietary software, contract law is essential for the propagation of the open source model.

**Software: Protection and Management**

Debate is raging as regards intellectual property and software. Suggestions vary from application to non-application of property rights to software; from correct interpretation, to amendment to outright replacement of property laws with alternative models of sui generis software protection. Fuelling this debate is the current success of open source software, arguably without the conventional mode of software protection.

As stated earlier, the open source faction staunchly opposes the current software patents regime. Essentially most open source concerns emanate from ineffective use and regulation of the patent system coupled with legal issues in context of software patents. A major bone of contention is the patent term which though considerably shorter than the copyright term is much more enveloping in effect due to its monopoly nature. However, though the potential for harm is great, it does not mean that ideas contained in software patents should not be protected. Patents in software are a relatively new development on the intellectual property front. As with other developments, the rough edges of this development too need to be smoothened out. Rather than completely eliminating patents as a medium of protection for software, efforts should be focused on a better system with higher standards of examination. Instead of complete radical substitution, mere supplementation and rectification of the patent system addressing these concerns is a possibility.

When open source supporters protest against software patents, an inevitable question that arises is whether copyright protection alone is adequate to protect computer software? Though the copyright regime has its fair share of supporters, critics, think-tanks and government bodies are largely of the view that copyright law by itself is an inadequate mode of protection for software and could actually be much worse for the industry and open source. US Judiciary too noted this deficiency. Primarily, copyright law by its very nature protects expressions and not the underlying ideas. In case of computer software, it transitions to protection of the expression but not the underlying functionality. Hence, the essence of computer software is open to analogous development. This in turn allows several expressions of the same innovation to exist. Some critics view this conflict as an opportunity to revisit copyright jurisprudence and redraw the lines to properly demarcate authorship issues, and resolve the idea-expression dichotomy, at least in context of software programs.

While some critics cite the need to revisit intellectual property jurisprudence, others challenge the very existence of property rights in software development. The open source movement is a viable mode of software development, and their argument that property rights should not apply to software may hold merit. However, the real question is whether absence of property rights would be incentive enough for software development. Moreover, a fundamental question to be answered is whether the open source movement would be an adequate replacement? The open source community may not be ideally suited to provide the requisite innovation impetus. Although many contributors appear to participate in open source software development for reputational rewards, it is unclear whether such intangible benefits are a
sufficiently effective motivation to drive large-scale efforts.\textsuperscript{31} More than initiation, sustaining of such projects is a big question. The debate has its share of supporters and detractors.\textsuperscript{63} Moreover, open source penetration is not sufficient and an overnight radical change could well mean disaster. Additionally, the software industry is not an island; any change in the protection regime, would have a cascading effect on several other industries. Furthermore, what needs to be considered is that if there is a complete replacement of the proprietary structure with the open source model, then the corresponding business model would alter the nature of the software industry, from that of software creation to software servicing.\textsuperscript{64}

It cannot be denied that a complementing stance needs to be acquired to accommodate the OSS model. The open source regime has acquired a stature which cannot be just brushed aside as an anomaly. Complete removal is not a good option; the open source model provides a good counterbalance in the software industry to the proprietary model. Besides preventing market dominance and monopoly it also allows for greater variety; companies are able to combine the best of both worlds.\textsuperscript{64} Also, regularizing the open source model allows its proper regulation. At the end of the day, it is not desirable to establish an exclusive system where only one of the policy models exists. Adjustments would be needed to harmoniously integrate the open source model with the proprietary model in the intellectual property regime. Any regime not doing so ‘runs the serious risk of remaining a vestige of twentieth-century IP regimes focused on rights of exclusion rather than enabling and encouraging rapid software development.’\textsuperscript{31}

Instead of focusing on the debate as a means, the focus should be on the ends. The need of the day is efficient and cost effective software. It is irrational, not to encourage any capable production mode, whether it be proprietary or open source. Both have their positive attributes and their shortcomings. Moreover, it is difficult to assess whether either model would be more successful without the influence of the other. What needs to be realized is that as things currently stand, both the modes cannot be done without. ‘Essentially, both the approaches require balancing of commerce and user independence. To completely dismiss one approach in favour of the other is to abandon a mature method of fostering innovation for another that is still immature.’\textsuperscript{67} Furthermore, courts and market inertia will force both the open source and proprietary models to coexist.\textsuperscript{39} ‘Given the uncertainties in determining which kinds of endeavours can be safely left to open innovation, it is likely that a dual system will be operative in many technological fields.’\textsuperscript{66} Hence, a harmonious construction of software protection is the best option.

**Conclusion**

Software programming tends to be based on de facto standards and the case of de facto standards is that they find favour only up to the time they are the best available solution. Earlier computer programming was done on one-to-one basis and cyberspace was yet to be established as a medium of connectivity. However, that is not the case anymore. Software is rarely individually tailored, but mass-produced, and the existence of the Internet makes global collaboration extremely easy. Furthermore, programming itself has become relatively easier. Not taking advantage of such a scenario would amount to restricting infrastructural growth. Open source development has become a practical alternative to more traditional proprietary production schemes for software.\textsuperscript{31} The open source faction is using the open source method to renegotiate and properly channelize electronic information governance.

There is not only intra-cooperation but also inter-cooperation with other development models. Open source shares an antagonistic relationship with proprietary software only as regards the mode of development. Otherwise, it is true that it can act as a substitute for proprietary software in certain cases, but it is also true that in other cases it makes the proper functioning of proprietary software possible.

Discussions as regards proper governance of ‘constructed commons’ have emerged lately.\textsuperscript{68-71} Most importantly, as private control and open access coexist on the Internet, their interplay needs to be properly regulated.\textsuperscript{72} To make open source a sound innovation development program requires concerted action from all involved stakeholders.

In view of increased government interest in OSS, the time is ripe for concrete actions to be taken to properly define the parameters of the OSS regime. Though the governments in various countries have set up committees, the main focus of such committees has primarily been on procurement and deployment.\textsuperscript{4,72} Other technical factors in software management which have ancillary effect on determination of improvements management in software should also be addressed.
Moreover, the governments need to promote development of open source software in their respective jurisdictions, especially amongst universities and technical institutes.

A complementing stance would have to be taken by the legislature too.\textsuperscript{66,69,73} Though litigation and private enforcement on its own accord is capable of resolving issues with copyright law, however, given the efficient management of resources that is required, legislative reform is preferable.\textsuperscript{5} Furthermore, the present times necessitate that certain guidelines be laid down addressing properly the treatment of OSS by IP laws. Some in the open source community may likely be ‘horrified’ at the idea that legislature might step in to try to regulate open source. However, the argument that open source practices are covered by IP laws and licensing practices and do not need any special catering to be countered. Legislative history is rife with instances where IP laws have been amended to address specific commercial issues, the popularly called Sonny Bono Act or the Mickey Mouse Protection Act in the US (The Copyright Term Extension Act, 1998) is a prominent example.

Given the popularity and wide-spread usage of OSS, decided litigation is relatively lacking in this field due in large part to pre-litigation negotiation, the high transaction costs associated with open source licensing enforcement, and the public policy pressure to respect open source programming rights.\textsuperscript{74} Still, the Judiciary has made some initial forays in the field of OSS. It is not just OSS, but certain concrete guidelines in general regarding software, need to be laid down, especially regarding derivative works in software and distribution and access to software both in the physical and the virtual medium.

Community involvement has been the leading light as regards formation and management of the open source philosophy. It turned into an extra-judicial arm and filled a perceived gap in software management which was not being addressed thus leading to the founding of OSS. However, despite the growing use and acceptance of OSS, the legal rights afforded to open source developers remain largely unknown.\textsuperscript{5} Furthermore, considering the dynamism of the software field, the involved community too needs to change its outlook and evolve with the times. The emergence of splinter factions within the open source philosophy bolsters this argument. It is true that the open source groups lobby against patent protection for software, but they need to take a more proactive role and lobby for overhaul of the software protection regime altogether.

The fact that the open source concept has evolved to the broader ‘open innovation’\textsuperscript{69} concept and the same model is being applied in other fields, prominently biotechnology, bioinformatics, genomics, and policy modelling, raises the issue that there might be several simmering grievances with the application of current IP laws to emerging fields. Hence, it would be desirable to review the IP system in the light of community development model as a whole.

Acknowledgment

Research for this paper was done at the Max Planck Institute for Intellectual Property & Competition Law, Munich, Germany and the Indian Law Institute, Delhi, India. The author expresses gratitude to the institutes for extending facilities for the research.

References

1 Raymond E, The Cathedral & The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary (O’Reilly Media, Massachusetts, USA), 1999.
5 Brown C, Copyleft, the disguised copyright: Why legislative copyright reform is superior to copyleft licenses, University of Missouri - Kansas City Law Review, 78 (2) (2010) 749.
7 Street F, Law of the Internet (Matthew Bender, NY), 2009, §14.05.
13 McManis C and Seo E, The interface of open source and proprietary agricultural innovation: Facilitated access and
16 Nichols v Universal Pictures Corp, 45 F.2d 119, 121 (2d Cir. 1930).
18 Computer Assocs Int'l v Altai Inc, 126 F.3d 365 (2d Cir. N.Y. 1997).
45 Georgiades E, Falling through the cracks: The problem of granting software patents, SCRIPTed, 7(3) (2010) 474.
52 Open invention network formed to promote linux and spur innovation globally through access to key patents (10 November 2005), http://www.openinventionnetwork.com/- press_release11_05.php (28 July 2010).