13 Multi-Licensing Model (Open and Closed Source) and Software Protection: Revisiting the Proposed Sui Generis Software Protection Models [(*)]

Invited Researcher: Vikrant Narayan Vasudeva [(**)]

There has been extensive discussion on the mode of intellectual property protection for software – copyright, patent, or sui generis. The sui generis approach essentially emphasize on attaining a balance between the rights granted by copyright and patent. Eventually, computer programs per se are protected by copyright, whereas technical applications of computer software are protected by patent law. The open source software licensing structure has emerged as a novel outlook towards copyright and patent law. The multi-licensing scheme utilized by certain corporate entities involving licensing of software under two structures- one of which is open source and the other "closed" source/"proprietary" has developed relatively recently. Studying the manner in which this multi-licensing scheme is applied reveals similarity to the suggestions made by the various sui generis models for software protection. Perhaps this is an indication of and validation that the software industry has evolved to such an extent that there needs to be a rethink of the current methods of software protection and how a better model could be developed.

Ⅰ Introduction

Since 1960s onwards and especially in the 1970s and 80s there has been extensive discussion on the mode of intellectual property protection for software – copyright, patent, or *sui generis*. The *sui generis* approaches essentially emphasize on attaining a balance between the rights granted by copyright and patent laws.

The industry relied primarily on trade secret law and contract law as the mode of software protection initially. Eventually, copyright law was chosen legislatively as the vehicle for software protection. Judicial recognition soon allowed patent protection as well. Soon industry started recognizing technological protection measures as another option and their efficacy was strengthened by legislative and judicial support.

Several occurrences in the past three decades highlight the developing fissures in the mode of software protection. Other developments highlight the increasing acceptance and ease in implementing *sui generis* protection.

Correspondingly, the open source software licensing structure has emerged as a novel outlook towards copyright law.[1] The multi-licensing scheme is utilized by certain corporate entities, where the software is licensed under two or more licensing structures- one of which is open source and the other "closed" source/ "proprietary." This allows the propagating entity to benefit from both the structures.

The manner of application of the multi-licensing scheme reveals a close approximate to the suggestions made by various studies supporting the *sui generis* models for software protection. Perhaps this is an indication that the software industry has evolved from the 70s and 80s to such an extent that there is a need to rethink the mode of protection accorded to software.

Ⅱ Legal Protection of Software

Software development initially was usually on an individual scale and not mass-produced. Trade secrets and individually tailored contracts were traditionally used to protect software. With the evolution and widespread usage of computer technology, individually tailored licenses gave way to mass-market software licenses. 1960s onwards, copyright law too began to be used to protect software.[2] 1970s saw discussions emerge as regards software patents[3] and *sui generis* proposals.

In the international arena, WIPO made *sui generis* proposals for software protection.[4] Several academic studies too emerged as regards software protection during this period, viz. *Galbi*,[5] *Menell*,[6] *Samuelson*,[7] *Stern,*[8] *Karjala*[9] and several other

academics.[10]

In Japan, The Ministry of International Trade and Industry (MITI) Study Committee on Legal Protection of Software issued an interim report in 1972 and found the copyright protection afforded to software as inadequate.[11] In 1973, the Second Subcommittee of the Copyright Council set up by the Agency for Cultural Affairs (*Bunka-cho*) submitted the ACA a report supporting copyright law as the mode of protection for software requiring minimal changes.[12] In 1983, Information Industry Committee, Industrial Structure Council set by MITI submitted an interim report that recommended a *sui generis* legislation to protect computer software- the Program Rights Law.[13] In 1984, the Sixth Subcommittee of the Copyright Council set up by the Agency for Cultural Affairs released an interim report recommending that Japan follow copyright law protection for software.[14] Due to heavy U.S. and European protest and lobbying, the Agency for Cultural Affairs' recommendation was followed.[15]

Eventually, software protection was explicitly brought under the ambit of copyright protection.[16] The 1970s saw emergence of patent law protection and by the 1990s they had increasingly become the favoured mode of software protection.[17] Later developments led to technological measures also protecting software. The possibility of reverse engineering to gain unauthorized access has been countered via law.[18]

International treatises, primarily The Trade Related aspects of Intellectual Property Rights (TRIPs) Agreement further fortified copyright law's position.[19] Some theorists are of the view that TRIPs also allows patentability of computer programs.[20] The WIPO Copyright Treaty has reinforced TRIPs position.[21]

With evolution of technology and software, and its increased adoption, usage and diversity of application, the calls for change have been becoming increasingly strident; several discussions as regards amendment to existing copyright law[22] and patent law[23] have been posited to better accommodate software. Simultaneously, discussions regarding *sui generis* protection have been becoming increasingly prominent in academia.[24] There have been dissents too against *sui generis* protection.[25]

Ⅲ Need of a Sui Generis Regime

Several occurrences in the past three decades highlight the developing fissures in the mode of software protection. Other developments highlight the increasing acceptance and ease in implementing *sui generis* protection.

1   The Software Industry

As personal computing evolved and technology became more accessible and affordable, programs have become increasingly user oriented and are increasingly permeating society. Emergence of the Internet phenomenon has fuelled software variety, proliferation and dispersion. It has also created new avenues for software applications to develop. Software's dual nature is increasingly becoming evident – it can be both tangible and intangible and sometimes the dual nature is simultaneously applied or revealed.[26] Software industry is largely based on "network effects;" this effects another mode of promoting reliance on a particular software and in a manner its indirect protection. User innovation has presented a new paradigm in software development; a prominent example being open source software.

2   Legal Protection of Software

Copyright law protects expressions and not the underlying ideas. For computer software, it transitions to protection of the expression but not the underlying functionality.[27] This allows several expressions of the same innovation to exist.

Software patents have been a dynamic field. From being unpatentable subject matter, to requiring an actual physical embodiment to reflect integration became a requisite for grant of protection. This slowly made way for a token physical embodiment and now a computer program on a disk too is considered to be patentable.[28] This dilution in stance grants recognition to the tangible/ intangible and functional/ industrial character of software which does not conform to the traditional notions of subject matter covered by patent law as against copyright law. The scenario has evolved to such an extent that *Mikus* argues that the physical embodiment criteria should be completely done away with.[29]

Currently layering of protection is being done where the components are protected by more than one set of protection options. Contract law is increasingly being used where software is now licensed instead of sold to avoid exhaustion and first sale liabilities.[30] Similarly, in addition to copyright law, trade secrets or technological measures are often used to protect software; these measures, in turn are further protected by

laws regulating and monitoring fair use, anti-circumvention, reverse engineering and interoperability. [31] Thus, layering of laws has become a "… tool that copyright holders have used to recalibrate the balance of rights among themselves, their users, and their competitors."[32] The need to have layering of laws questions the efficacy of the existing regime for software protection.[33] Overlapping protection and layering of laws also increases transaction costs, where investment has to be made in each form of protection separately.

3 Evolving Landscape of Technology-Law Interaction

Information products are increasingly becoming hybrid where distinguishing between the real and virtual is becoming complicated. Traditional roles of hardware and software are increasingly becoming functionally interchangeable. This complicates traditional notions of application of intellectual property laws. Prominent examples of the complexity and dichotomy arising out of technological integration are online storage, cloud computing, interface protection, and business method patents.

4 Legal Recognition of Protection for Evolving Technological Arenas

The past decades have seen legal recognition being accorded to several arenas that would ordinarily have been attempted to be accommodated under the traditional intellectual property laws. "… several 'carve out' statutes have been enacted for the special protection of certain intellectual property that shares both functional and expressive characteristics." [34] Prominent examples are Utility Model laws, [35] Design protection, [36] Semiconductor Chip protection, [37] Vessel Hull Design protection, [38] and Database protection. [39] Even within copyright and patent laws greater clarity has been attempted to be achieved as regards computer software. This can be seen by virtue of the EU Directive on Legal Protection of Computer Programs [40] and debate on EU Proposal for a Directive on the Patentability of Computer-Implemented Inventions. [41] These have furthered the arguments to recognize *sui generis* protection for software.[42]

5 International Harmonization of Laws

The present scenario for building international consensus is relatively more conducive. Several attempts at achieving international harmonization are being and have already been made. The TRIPs Agreement is a prominent example. U.S. has also "added business method patenting to the international intellectual property harmonization agenda."[43] The TRIPs Plus arrangement is being promoted too, which creates bilateral agreement channels for negotiation.

Ⅳ The Multi-Licensing Approach

As opposed to the generically referred "conventional," "proprietary," "closed," "hidden," "restrictive" source code model, the open source software model emphasizes on unrestricted accessibility to the source code of the program and a community based development model. Open source does not simply mean freedom of access to source code. It includes much more depending upon the standard setting body spearheading the project.[44]

The open source software community essentially employs two polar outlooks towards the treatment of improvements – one where the developed code and improvements are donated in general under a license which allows complete access to the code even allowing it to be made proprietary. The second option and more important from a legal point of view is where the improvements developed need to be licensed under the terms of the same license as the initial code was. This allows retention of control over improvements by the licensor. The former are called Academic licenses while the latter, Reciprocal licenses.

The reciprocal obligation is known by various informal terms, viz. 'Viral', 'Taint', 'Infectious', and 'Copyleft.' Copyleft is a way of using of the copyright on the program. It is a general concept, one of whose specific and most prominent implementation is in GNU GPL.[45]

The multi-licensing scheme is utilized by certain corporate entities,[46] where the software is licensed under two or more licensing structures- one of which is open source and the other "closed" source/ "proprietary." This allows the propagating entity to benefit from both the structures. "[T]he dual licensor offers bifurcated terms, and the distributor-licensee chooses to operate on one side of the bifurcation or the other. The originating dual licensor, however, can incorporate software revisions it finds on the open source side into the proprietary side."[47]

Ⅴ Comparative Analysis of Sui Generis Studies for Software Protection with the Multi-Licensing (Open Source-Closed Source) Approach and Proposal for a Model Software Law

Perhaps the MITI (Information Industry Committee, Industrial Structure Council) proposal (hereinafter, simply called "MITI" or "MITI proposal") proposal[48] was a bit ahead of the times and not the most frictionless policy capable of implementation in the 1980's. However, the manner in which the software field has developed indicates that most of its proposals are extremely relevant currently.

A comparison of the MITI and other *sui generis* proposals with the multi-licensing approach indicates a number of similarities. The multi-licensing approach seems to validate the hypothesis of the *sui generis* proposals to a considerable extent.

1 Conditions for Grant of Protection

Copyright law requires originality, while patent law requires novelty, utility and non-obviousness for grant of protection. *Stern* proposes an amalgamation of the criteria for software as "… originality, novelty, and technical advance."[49] Additionally, the criteria of utility and sweat of the brow[50] too need to be considered. The originality-novelty balance should ideally apply in such a manner that neither cloning nor independent creation of a substantially similar product would be allowed. Copying should not be restricted to literal copying, but as under patent law, any obvious equivalents too should be restricted. [51] GPL's reciprocity obligation [52] controls any "work based on the program,"[53] or any modification[54] thus constituting a "covered work." [55] Transposing this analogy to a legal regime, 'sweat of the brow' doctrine could be accommodated. Possibly, the term of protection granted to elements covered by the 'sweat of the brow' doctrine could be less. Furthermore, categorization of stringency of criteria for grant along a range could be developed.

2 Scope of Protection: General

Demarcating the parameters of software is difficult and furthered by the fact that the *sui generis* model would need to accommodate both the fields of copyright and patent laws.[56] MITI in 1983 rightly conceptualized that "software includes everything related to computer applications [source programs and object programs … flowcharts, manuals, ideas, algorithms, etc.]." [57] However, as regards the subject matter of protection, *MITI* stated that "Programs (source programs and object programs) are subject to the protection."[58] It is preferable to have a general broad classification similar to the one suggested by MITI. However, the subject matter of protection, instead of just being source code and object code should be extended to the complete arena of computer software, thus encompassing source code, object code, algorithms, manuals, flowcharts, methods of doing business etc.[59] Ideas per se though should be excluded.

3 Scope of Protection: Incremental Innovation

A considerable amount of software improvements may not be eligible for protection under the traditional intellectual property regime.[60] Also, the nature of software industry and current laws tend to closet building blocks. "A market-oriented legal regime should avoid wasteful reduplications of effort."[61] Open source approach, via its copyleft concept has adopted a unique method of managing incremental innovation which could be a workable option for the software industry. Analogously, the contributors would also have a stake in the revenue generated from the parent software. The *Manifesto* advises use of blocking periods, during which the rights granted would be considerable curtailed. [62] Furthermore, it advises that such blocking periods could selectively distinguish between markets.[63]

4 Duration of Protection

Sui generis studies consider the term of protection under copyright and patent laws to be inordinately long in context of software, considering its utilitarian character,[64] short innovation and life cycle, [65] and non-access to 'building blocks.' It needs to be considered that just recouping research and development costs is not sufficient in the field of software. Software as an industry needs to build considerably on preexisting code. If the code already developed is taken into the public domain after every 2-5 years then there is constant whittling away at the base on which a software company rests. Also, later versions of a software incorporate a considerable amount of

code from earlier versions. Hence, a short innovation cycle cannot be an excuse for completely revoking protection for an earlier version as soon as the next version becomes available. The multi licensing scheme, essentially favours a long term of protection with diluted rights for one channel and strict rights for the second channel. Comparing with the multi licensing scheme but considering the other leeway allowed under the proposal, especially the almost immediate protection and compulsory licensing, the term of protection could perhaps be thirty years. Perhaps in the future this period of protection could be reduced.

5   Grant of Rights

MITI proposed the grant of right of use, alteration (limited extent), reproduction and lending a computer program.[66] Almost analogously, GPLv2 grants the rights to copy, distribute and modify.[67] GPLv2 and v3 though do not mention a use right, mention the right to run.[68] *Rosen* is of the view that right to run might be actually a right to use.[69] GPLv3 also includes a Patent clause which grants as far as any contributions are concerned most of the rights governed by patent law.[70] Though GPLv3 does not grant a have made right in its patent grant, it addresses the situation in the license.[71] Hence, under the proposed Model Software Law the owner would have the exclusive rights to do and to authorize the rights to use, reproduce, modify, make, have made and lease.

6   Moral Rights

MITI report advised against recognizing moral rights in computer programs,[72] primarily because of its commercial nature;[73] the author by virtue of moral rights should not be able to control whether and when and in what form the work should be made public.[74] Furthermore, the technical nature of software reduces the cultural foundation of moral rights. MITI advises that "…where it is necessary to approve a moral right, protection by the general principles of the Civil Code is considered to be sufficient"[75] The open source approach implements in a limited manner the spirit of moral rights; both GPLv2[76] and GPLv3[77] recognize the right of attribution subject to contractual provisions. Considering the successful utilization of the contractually controlled right of attribution by the open source approach to attract developers based on

reputational reward, limited attribution right could be granted under the Model Software Law.

7   Limitations on Exclusive Rights

Several usually recognized exceptions under copyright law are also proposed under the Model Software Law, viz. archival/ backup copy, copy for maintenance or repair, Scenes a Faire, de minimis use, equitable defenses and public domain elements. As only a right of lease and not sale is being provided,[78] application of the exhaustion doctrine would be extremely limited. Fair use would also be recognized. GPLv3 explicitly states this exception too.[79] Reverse Engineering of software has lately been an extremely tumulus arena. To promote innovation, interoperability should be encouraged.[80] Besides the "copyleft" reciprocal obligation, GNU GPLv3 contains a specific clause limiting usage of technological protection measures.[81] It is preferable that the Model Software Law involves a mechanism where an entity seeking interoperability be provided all facilities by the owner of the original code so that there should not exist any need to reverse engineer at all. Such a right should differentiate between private and commercial use. For commercial use, the right should follow the licensing mechanism of being subject to blocking periods and adequate royalty payments.[82] *Stern* suggests debugging as an exception too.[83] However, though it may be excepted for self use, upon distribution, the same rules as under incremental innovation and licensing would be applicable. Innocent infringement too needs to be considered. GPLv3 makes provision for reinstating of license both temporarily and permanently hence providing solace to inadvertent violators.[84]

8   Negotiated Licenses and Compulsory Licensing

"[L]arge firms stifle competition by reserving broad markets of algorithms, then refuse to license (or demand high license fees) or force competitors to waste resources inventing less efficient noninfringing workarounds."[85] Hence, *sui generis* studies favour compulsory royalty bearing licensing for software[86] ensuring a constant revenue stream,[87] avoiding duplication, thus ensuring efficiency[88] and securing incremental innovation.[89] GPL's "copyleft" reciprocal obligation clause is a type of automatic compulsory licensing provision. "The compulsory licensing provision would require third parties wishing to license the

rights holder's technology first attempt to negotiate a license with the rights holder on 'reasonable commercial terms and conditions.'"[90] Compulsory licensing would involve an analysis of the market dynamics. The status of the licensee- whether commercial or amateur and the reason for the license- whether commercial use or personal use. There could also be an option of a reciprocal obligation as exists under open source licensing or the option of a Right of First Refusal. Collecting societies could manage compulsory licensing issues.[91] Currently too, several open source groups manage licensing in a fashion akin to collecting societies.

9   Infringement

Software is unique in that it involves both abstract and literal aspects, stock components and unique components, and a blend of artistic, aesthetics and functional. Each aspect needs to be approached differently to define the infringement criteria. Observing the jurisprudence in copyright and patent laws highlights the difficulties of such an endeavour. A major difference between infringement analysis under patent law and copyright law is independent creation. Given the balance being attempted to be achieved under the proposed Model Software Law, "[t]hat the defendant did not actually copy the advance and independently developed the infringing product probably should not be a defense."[92] *Soderquist's* proposal of applying the doctrine of equivalents to software should probably be followed.[93] *Samuelson et. al.* developed a considerable thesis to highlight the similarity dimension regarding two products[94] Varying determinative criteria should be applied to determine infringement of the abstract components as against those of the literal components. As regards the abstract components, 'use by embodiment' and 'trafficking in embodiments' have been suggested as constituting infringement.[95] The merger doctrine should not be applicable under the Model Software Law.[96] Also, varied levels of infringement should apply to the various software components depending on social utility, aesthetics and ease in determining infringement.[97]

10   Remedies

Injunctions are a usual remedy in infringement cases. However, in context of software depending on when the infringement is detected and the level of subsequent development on that particular software, completely preventing usage of the infringing software might be harmful to the industry. Hence, another remedy of allowing usage subject to royalty payments for both past and future use would have to be made available. Such a royalty would need to imbibe a penalty effect and be higher than that which would have been available under the compulsory licensing scheme. The level of the royalty scheme could be based on a determination whether the infringement was willful or innocent. GPLv3 makes provision for reinstating of license both temporarily and permanently, hence providing solace to inadvertent violators.[98] The MITI proposal provides the remedies of "injunction, measures to restore business reputation, estimation of damages and penal provisions"[99] However, "[i]t is questionable whether criminal penalties are appropriate here."[100] Impounding and disposition of infringing articles would also be available.

11   Dispute Resolution

Need of technical knowledge in the legal experts, maintaining confidentiality, and the short lives of software are some of the motivating factors according to *MITI* which prompt a simple, fast and fair means for dispute settlement as regards computer software.[101] Hence, the *MITI* report favours "a mediation, conciliation, arbitration and judgment system"[102] instead of adversarial court procedures.[103] Most cases of open source code are decided through private settlement. Considering the amount of open source code in use and the duration for which it has been in use, litigation in court has been sparse and decided litigation even less. This shows that private settlements are an effective software dispute settlement procedure.

12   Administrative Procedure

Neither the instantaneous according of protection under copyright law, nor the elaborate procedures under patent law are suitable for software. As regards the Model Software Law, instead of defining the claims, it is preferable that a deposit of the complete code be done, and as soon as the deposit and a basic form is submitted, registration be accorded. Provision for keeping the code secret could also be made or an escrow procedure could be established. "The right would thus be much easier to obtain than a standard patent but not automatically granted like

copyright."[104]

Complementing this stance, time consuming administrative procedures like examination should be postponed till an infringement action is brought, whereupon such determinations would be made.[105] Such a provision should be especially secured against frivolous challenges. This coupled with the Peer-To-Patent (P2P) procedure could ease the examination procedure immensely.[106]

VI Conclusion

This research has attempted to make a case for the need of a *sui generis* model for software protection and what should be its form. Computer software's categorization under copyright law and outside industrial property shifts the focus of the laws on to the individual, while the demands of the industry require it to be recognized as industrial property; this creates a degree of friction. The proposed Model Software Law in this research is primarily based on the recognition that software tends to fall more in the arena of industrial property than copyright law. This research bases the *sui generis* approach on the multi-licensing scheme. Firstly, a multi-licensing model tries to establish the relevant balance between open source software and closed-source software, reaping the benefits of both the approaches. It is preferable that any innovation promoting *sui generis* scheme does the same. Secondly, this research submits that the multi-licensing model is a close approximation to several proposed *sui generis* models of software protection, which too have aimed to balance the interests between patent and copyright laws. Lastly, observing from the practice of the multi-licensing scheme, it would allow greater predictability of how a *sui generis* software protection model would function. The success of such a *sui generis* scheme would significantly rest on building consensus. Opportunely, successes of achieved and attempted harmonization discussions have established channels which can be availed of in accomplishing a consensus.

---

1 In this research, the term open source has been used in its generic-industry recognized form encompassing the specific terms "Open Source software," "Free Software," "Libre Software" and other similar software in the category. However, primarily, the focus of this research is on the GNU General Public License (GPL) propagated by the Free Software Foundation.

2 See generally, Jack M. Haynes, *Computer Software: Intellectual Property Protection in The United States and Japan*, 13 J. Marshall J. Computer & Info. L. 245 (Winter, 1995).

3 See generally, NOBUHIRO NAKAYAMA, LEGAL PROTECTION OF SOFTWARE (Sofutowea no hoteki hogo), 9-14 (New ed. 1988). Ch. 1, Sect. 2, Legislative History.

4 See, Model Provisions on the Protection of Computer Software, WIPO No.814 (E) (1978) [Hereinafter, WIPO Model Provisions, 1978]; Draft Treaty for the Protection of Computer Software, WIPO, LPCS/II/3 (1983) [Hereinafter, WIPO Draft Treaty, 1983].

5 Galbi, *Proposal for New Legislation to Protect Computer Programming*, 17 (4) Bulletin of the Copyright Society of the U.S.A. 280 (1970); Galbi, *The Prospect of Future Legislation and Court Action Concerning the Protection of Programming*, 13 Jurimetrics Journal 234 (Summer 1973).

6 Peter S. Menell, *Tailoring Legal Protection for Computer Software*, 39 Stan. L. Rev. 1329 (July, 1987); Peter S. Menell, *The Challenges of Reforming Intellectual Property Protection for Computer Software*, 94 Colum. L. Rev. 2644 (1994).

7 Pamela Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Readable Form*, 1984 Duke L.J. 663 (1984); Pamela Samuelson, *Creating A New Kind of Intellectual Property: Applying the Lessons of The Chip Law to Computer Programs*, 70 Minn. L. Rev. 471 (December, 1985); Pamela Samuelson, *Benson Revisited: The Case Against Patent Protection for Algorithms and other Computer Program-Related Inventions*, 39 Emory L.J. 1025 (Fall, 1990); Pamela Samuelson, Randall Davis, Mitchell D. Kapor, & J.H. Reichman, *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 Colum. L. Rev. 2308 (December, 1994).

8 Richard H. Stern, *The Bundle of Rights Suited to New Technology*, 47 U. Pitt. L. Rev. 1229 (Summer, 1986); Richard H. Stern, *A Sui Generis Utility Model Law as An Alternative Legal Model for Protecting Software*, 1 U. Balt. Intell. Prop. L.J. 108 (Spring, 1993); Richard H. Stern, *Solving The Algorithm Conundrum: After 1994 in The Federal Circuit Patent Law Needs A Radical Algorithmectomy*, 22 AIPLA Q.J. 167 (Spring, 1994); Richard H. Stern, *On Defining The Concept of Infringement of Intellectual Property Rights in Algorithms and other Abstract Computer-Related Ideas*, 23 AIPLA Q. J. 401 (1995).

9 Dennis S. Karjala, *Lessons from the Computer Software Protection Debate in Japan*, 1984 Ariz. St. L.J. 53 (1984); Dennis S. Karjala, *The Limitations on the Protection of Program Works Under Japanese Copyright Law*, 8 Mich. YBI Legal Stud. 25 (1987); Dennis S. Karjala, *Copyright, Computer Software, and The New Protectionism*, 28 Jurimetrics Journal 33 (Fall 1987); Dennis S. Karjala, *The Protection of Operating Software under Japanese Copyright Law*, 29 Jurimetrics Journal 43 (Fall 1988); Dennis S. Karjala, *A Coherent Theory for The Copyright Protection of Computer Software and Recent Judicial Interpretations*, 66 U. Cin. L. Rev. 53 (Fall, 1997).

10 Duncan M. Davidson, *The Future of Software Protection: Common Law, Uncommon Software*, 47 U.

Pitt. L. Rev. 1037 (Summer, 1986); Max W. Laun, *Improving the International Framework for the Protection of Computer Software*, 48 U. Pitt. L. Rev. 1151 (Summer, 1987); Rochelle Cooper Dreyfuss, *Information Products: A Challenge to Intellectual Property Theory*, 20 N.Y.U. J. Int'l L. & Pol. 897 (Summer, 1988); Tohru Nakajima, *Legal Protection of Computer Programs in Japan: The Conflict Between Economic and Artistic Goals*, 27 Colum. J. Transnat'l L. 143 (1988-1989); Glynn S. Lunney, Jr., *Copyright Protection for ASIC Gate Configurations: PLDS, Custom and Semicustom Chips*, 42 Stan. L. Rev. 163 (November, 1989); John C. Phillips, *Sui Generis Intellectual Property Protection for Computer Software*, 60 Geo. Wash. L. Rev. 997, 1012 (1992); A. Samuel Oddi, *An Uneasier Case for Copyright than for Patent Protection of Computer Programs*, 72 Neb. L. Rev. 351 (1993); Kenneth W. Dam, *Some Economic Considerations in the Intellectual Property Protection of Software*, 24 J. Legal Stud. 321 (June, 1995).

[11] See, Kenneth L. Port and Gerald Paul McAlinn, Comparative Law: Law and the Legal Process in Japan, 670 (Carolina Academic Press) (2nd Ed. 2003) *citing* Welch and Anderson, *Copyright Protection of Computer Software* in Japan 11 Computer J.L. 287 (1991).

[12] See generally, Nakayama, *supra* n. 3.

[13] Id.

[14] Nakayama, *supra* n. 3.

[15] Id.; Also see, Haynes, *supra* n. 3.

[16] See, U.S. Copyright Act 17 U.S.C. § 101 & § 102 (1976); (U.S.A.) H.R. Rep. No. 96-1307, Part 2, 96th Cong. 2d Sess. 19 (1980); (U.S.A.) H.R. Rep. No. 94-1476, Cong., 2d Sess. 54 (1976); S.31, (Australia) Copyright Act, 1968; Art. 10(1) (ix), Japan Copyright Law (1970); European Union, Council Directive 91/250/EEC Legal Protection of Computer Programs, OJ L 122, (May 17, 1991).

[17] See, United States Patent and Trademark Office (USPTO), The Manual of Patent Examining Procedures § 2106; USPTO Examination Guidelines for Computer-Implemented Inventions, 61 Fed. Reg. 7478 (1996); Arts. 52(2) & 52(3), European Patent Convention (1973); European Patent Office (EPO), Guidelines for Examination in the European Patent Office (2001); Japan Patent Office (JPO), Examination Guidelines for Computer Software Related Inventions (1993); JPO, Examination Guidelines for Computer Software Related Inventions (1997); JPO, Implementation Guidelines for Examinations in Specific Fields, Chapter 1- Computer Software-related Inventions (1999); JPO, Examination Guidelines for Computer Software-related Inventions (2000)

[18] Arts. 11 & 12, WIPO Copyright Treaty (1996); U.S. Digital Millennium Copyright Act 17 U.S.C. § 1201 (1998); European Directive 2001/29/EC Harmonisation of Certain Aspects of Copyright and Related Rights in The Information Society (2001).

[19] Art.10 (1), Agreement on Trade-Related Aspects of Intellectual Property Rights (1995).

[20] See, Eloise Gratton, *Should Patent Protection be Considered for Computer Software-Related Innovations?*, 7 Comp. L. Rev. & Tech. J. 223 (Winter, 2003).

[21] Arts. 4, 11 and 12, WIPO Copyright Treaty (1996).

[22] See, Kristina Soderquist, *Yet Another Suggestion for Solving the Computer Program Dilemma*, 6 Va. J.L. & Tech. 14 (Fall, 2001); Catherine Parrish, *Unilateral Refusals to License Software: Limitations on the Right to Exclude and the Need for Compulsory Licensing*, 68 Brooklyn L. Rev. 557 (Winter, 2002); Susan Corbett, *What Ifs and other Alternative Intellectual Property and Cyberlaw Story: What If Object Code Had Been Excluded from Protection as A Literary Work in Copyright Law? A New Zealand Perspective*, 2008 Mich. St. L. Rev. 173 (Spring, 2008).

[23] See, Vincent Chiappetta, *Patentability of Computer Software Instruction as an "Article Of Manufacture:" Software as Such as The Right Stuff*, 17 J. Marshall J. Computer & Info. L. 89 (Fall, 1998); Mark Aaron Paley, *A Model Software Petite Patent Act*, 12 Santa Clara Computer & High Tech. L.J. 301 (August, 1996); Chad King, *Abort, Retry, Fail: Protection for Software-Related Inventions in The Wake of State Street Bank & Trust Co. v. Signature Financial Group, Inc.*, 85 Cornell L. Rev. 1118 (May, 2000); Richard S. Gruner, *Intangible Inventions: Patentable Subject Matter for an Information Age*, 35 Loy. L.A. L. Rev. 355 (January, 2002); Gratton, *supra* n. 20.

[24] Glynn S. Lunney, Jr., *E-Commerce and Equivalence: Defining the Proper Scope of Internet Patents*, 7 Mich. Telecomm. & Tech. L. Rev. 363 (2000-2001); Toshiko Takenaka, *E-Commerce and Equivalence: Defining the Proper Scope of Internet Patents Symposium: Commentary: International and Comparative Law Perspectives on Internet Patents,* 7 Mich. Telecomm. Tech. L. Rev. 423 (2000 / 2001); Vincent Chiappetta, *Defining the Proper Scope of Internet Patents: If We Don't Know Where We Want to Go, We're Unlikely to Get There*, 7 Mich. Telecomm. & Tech. L. Rev. 289 (2000-2001); Jean−Paul Smets−Solanes, *Patent or Sui Generis Right: What Protection Should Be Considered for Software and other Intangible Innovations?,* (2001); Scott J. Hill, *Restructuring The Law: Proposing A New Section of Title 35 and The Effect Of Amazon.Com v. Barnesandnoble.Com on Business Method Patents*, 48 Wayne L. Rev. 1239 (Fall, 2002); Bruce Abramson, *Promoting Innovation in The Software Industry: A First Principles Approach to Intellectual Property Reform*, 8 B.U. J. SCI. & TECH. L. 75 (Winter, 2002); Aaron D. Charfoos, *How Far Have We Come, and Where Do We Go From Here: The Status Of Global Computer Software Protection Under The TRIPS Agreement*, 22 NW. J. Int'l L. & Bus. 261 (Winter, 2002); Richard S. Gruner, *Everything Old is New Again: Obviousness Limitations on Patenting Computer Updates Of Old Designs*, 9 B.U. J. Sci. & Tech. L. 209 (Summer 2003); Vincent Chiappetta, *Trip-Ping Over Business Method Patents*, 37 Vand. J. Transnat'l L. 181 (January, 2004); Allen Clark Zoracki, *When is an Algorithm Invented? The Need For A New Paradigm For Evaluating an Algorithm for Intellectual Property Protection*, 15 Alb. L.J. Sci. & Tech. 579 (2005); Steven B. Toeniskoetter, *Protection of Software Intellectual Property in Europe: An Alternative Sui Generis Approach*, 10(2) University of San Francisco Intellectual Property Law Bulletin (2005); Matt Flinders, *Protecting Computer Software--Analysis and Proposed Alternative*, 7 J. High Tech. L. 172 (2007); Katherine J. Strandburg, *What if*

There Were A Business Method Use Exemption to Patent Infringement?, 2008 Mich. St. L. Rev. 245 (Spring, 2008).

25 Jane C. Ginsburg, *Four Reasons and A Paradox: The Manifest Superiority Of Copyright over Sui Generis Protection of Computer Software*, 94 Colum. L. Rev. 2559 (Dec, 1994); John M. Griem, Jr., *Against A Sui Generis System of Intellectual Property for Computer Software*, 22 Hofstra L. Rev. 145 (1993-1994); Paul Goldstein, *Comments on A Manifesto Concerning The Legal Protection of Computer Programs*, 94 Colum. L. Rev. 2573 (1994).

26 Davidson, *supra* n. 10 at 1072.

27 Abramson, *supra* n. 24 at 123; Karjala, *Computer Software, and the New Protectionism, supra* n. 9 at 94; Karjala, *A Coherent Theory, supra* n. 9 at 57.

28 See, footnote n. 17.

29 Jean-Philippe Mikus, *Of Industrious Authors and Artful Inventors: Industrial Works and Software at the Frontier of Copyright and Patent Law*, 18 I.P.J. 187, 253 (Aug 2004).

30 Stacey L. Dogan and Joseph P. Liu, *Copyright Law and Subject Matter Specificity: The Case of Computer Software*, 61 N.Y.U. Ann. Surv. Am. L. 203, 230 (2005).

31 Eg. WIPO Copyright Treaty; 17 U.S.C. § 1201 The Digital Millennium Copyright Act, 1998.

32 Dogan and Liu, *supra* n. 30 at 228.

33 Samuelson, *Creating A New Kind Of Intellectual Property, supra* n. 7 at 515; also see Gratton, *supra* n. 20 at 248.

34 Flinders, *supra* n. 24 at 195.

35 Art. 1, Paris Convention for the Protection of Industrial Property (WIPO, 1883); (Germany) Utility Model Law, 1986; (Japan) Utility Model Act (Act No. 123 of 1959); (Korea) Utility Model Law (No. 952 of 1961).

36 Art. 25, Agreement on Trade-Related Aspects of Intellectual Property Rights (1995); (U.S.) Patent Act, 35 U.S.C. § 171 (2000); EU Directive on the Legal Protection of Designs 98/71/EC (1998).

37 See, Washington Treaty on Intellectual Property in Respect of Integrated Circuits (1989); Art. 35, Agreement on Trade-Related Aspects of Intellectual Property Rights (1995); U.S. Semiconductor Chip Protection Act of 1984 Act of Nov. 8, 1984, Pub. L. 98-620, 98 Stat. 3347 (17 U.S.C. §§ 901-14); EU Directive on The Legal Protection of Topographies of Semiconductor Products (87/54/EEC 16 December 1986).

38 U.S. Vessel Hull Design Protection Act (Act of Oct. 28, 1998), Pub. L. 105-304, 112 Stat. 2860 (17 U.S.C. §§ 1301-1332).

39 Art. 10, Agreement on Trade-Related Aspects of Intellectual Property Rights (1995); EU Directive on the Legal Protection of Databases 96/9/EC (1996).

40 EU Directive on the Legal Protection of Computer Programs (91/250/EEC) (1991).

41 The European Commission, Proposal for Directive on the Patentability of Computer-Implemented Invention (CII Directive -2002/ 0047/COD) (2002).

42 Zoracki, *supra* n. 24 at 604; Flinders, *supra* n. 24 at 195-196.

43 See generally, Chiappetta, *Trip-Ping, supra* n. 24.

44 See, United Nations Development Programme, *The International Open Source Network (IOSN)*, available at http http://www.apdip.net/projects/2003/iosn (Last visited on October 17, 2011) (Free/Open Source Software (FOSS) is software that can be used, copied, studied, modified and redistributed without restriction); Also See, World Intellectual Property Organization, *Open Source*, available at http://www.wipo.int/patent-law/en/developments/open_source.html (Last visited on October 17, 2011) (Generally speaking, open-source software refers to software for which the source code (underlying programming code) is made freely available for use, reading the code, changing it or developing further versions of the software, including adding amendments to it. Open source software is often used as a general expression for many forms of non-proprietary software, which differ principally in respect of the licensing terms under which changed versions of the source code may be further distributed).

45 S.2(b) GNU GPLv2 (You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work … provided that you … cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole … under the terms of this License); S.5(c) of GPLv3 (You may convey a work based on the Program, or the modifications to produce it from the Program … provided that … You must license the entire work, as a whole, under this License … This License will therefore apply … to the whole of the work, and all its parts, regardless of how they are packaged).

46 See, http://www.oss-watch.ac.uk/resources/duallicence2.xml (last visited on October 26, 2011) (MySQL AB's database; Qt, a cross platform toolkit used to develop GUIs, from Nokia (originally Trolltech); Berkeley DB, a database system, from Oracle (originally Sleepycat software); Asterisk, an open source telecommunications software suite, from Digium).

47 Greg R. Vetter, *Exit and Voice in Free and Open Source Software Licensing: Moderating the Rein over Software Users*, 85 Or. L. Rev. 183, 224 (2006).

48 See, Ministry of International Trade and Industry, Industrial Structure Council Information Industry Committee, Interim Report, Dec. 1983 [Hereinafter, The MITI Report], reprinted in Nobuhiro Nakayama, Legal Protection of Software (Sofutowea no hoteki hogo), 214 app. 2.

49 Stern, *The Algorithm Conundrum, supra* n. 8 at 218.

50 Samuelson et al., *Manifesto, supra* n. 7 at 2355.

51 Cf. WIPO Model Provisions, 1978 *supra* n. 4.

52 S.2(b) GNU GPLv2; S.5(c) GNU GPLv3.

53 S.0, GNU GPLv2.

54 S.0, GNU GPLv3.

55 S.0, GNU GPLv3.

56 Oddi, *supra* n. 10 at 447 (… [I]t is far from clear that a sui generis system would provide bright lines at the interfaces of patent and copyright, and this system may indeed complicate an already unclear situation with respect to the interrelationships of these titles of protection.).

● 9 ●

57 The MITI Report, *supra* n. 48 at 221.

58 Id at 226. (Chapter V *proposal for MITI Program Rights Act,* Outline(2)).

59 See, Paley, *supra* n. 23 at 377; Stern, *The Algorithm Conundrum*, *supra* n. 8 at 213; Stern, *Defining The Concept of Infringement*, *supra* n. 8 at 519 (1995); Stern, *Sui Generis Utility Model Law*, *supra* n. 8 at 112.

60 Samuelson et al., *Manifesto, supra* n. 7 at 2365.

61 Id. at 2409.

62 Id. at 2371.

63 Id.

64 Samuelson, *CONTU Revisited*, *supra* n. 7 at 766.

65 Paley, *supra* n. 23 at 317; Toeniskoetter, *supra* n. 24 at 78; Zoracki, *supra* n. 24 at 595; Corbett, *supra* n. 22 at 181; Samuelson, *CONTU Revisited*, *supra* n. 7 at 766; Stern, *Bundle of Rights*, *supra* n. 8 at 1250.

66 The MITI Report, *supra* n. 48 at 226 (Chapter V, *Proposal for MITI Program Rights Act, Outline (3)*); also see, Paley, *supra* n. 23 at 374; WIPO Model Provisions, 1978 *supra* n. 4 (S.5 Rights of the Proprietor); Art. 4, WIPO Draft Treaty, 1983 *supra* n. 4.

67 S.0, GNU GPLv2; S.1, GNU GPLv2; S.2, GNU GPLv2.

68 S.0, Paragraph 2, GNU GPLv2; S.2, GNU GPLv3.

69 See, LAWRENCE ROSEN, OPEN SOURCE LICENSING: SOFTWARE FREEDOM AND INTELLECTUAL PROPERTY LAW, 225 (2005) available at http://www.rosenlaw.com/oslbook.htm (Last visited on April 25, 2011).

70 S.11, GNU GPLv3.

71 S.2, Paragraph 2, GNU GPLv3.

72 The MITI Report, *supra* n. 48 at 222.

73 Id.; Karjala, *Software Protection Debate in Japan, supra* n. 9 at 66.

74 Karjala, *Software Protection Debate in Japan, supra* n. 9 at 64; Max W. Laun, *Improving The International Framework for The Protection of Computer Software*, 48 U. Pitt. L. Rev. 1151, 1177 (Summer, 1987).

75 The MITI Report, *supra* n. 48 at 222.

76 S.1, GNU GPLv2; S.2(a), GNU GPLv2.

77 S.5(a), GNU GPLv3; S.7(c), GNU GPLv3.

78 Cf. Paley, *supra* n. 23 at 347.

79 S.2, Paragraph 1, GNU GPLv3.

80 Samuelson, *Creating A New Kind of Intellectual Property*, *supra* n. 7 at 524; But see, Stern, *Bundle of Rights*, *supra* n. 8 at 1252.

81 S.3, GNU GPLv3.

82 Samuelson et al., *Manifesto, supra* n. 7 at 2404; Paley, *supra* n. 23 at 344.

83 Stern, *Defining The Concept of Infringement*, *supra* n. 8 at 484.

84 S.8, GNU GPLv3.

85 Paley, *supra* n. 23 at 340 citing John Swinson, *Copyright or Patent or Both: An Algorithmic Approach to Computer Software Protection*, 5 Harv. J. L. & Tech. 145 at 168-69 (1991).

86 The MITI Report, *supra* n. 48 at 277 (Chapter V *Proposal for MITI Program Rights Act, Outline(8)*); Samuelson et al., *Manifesto, supra* n. 7 at 2371; Toeniskoetter, *supra* n. 24 at 77; Paley, *supra* n. 23 at 378.

87 Samuelson et al., *Manifesto, supra* n. 7 at 2414.

88 Id. at 2418; Karjala, *Software Protection Debate in Japan*, *supra* n. 9 at 68.

89 Samuelson et al., *Manifesto, supra* n. 7 at 2415; Toeniskoetter, *supra* n. 24 at 79.

90 Toeniskoetter, *supra* n. 24 at 77; also see, Paley, *supra* n. 23 at 378.

91 Paley, *supra* n. 23 at 341.

92 Stern, *Bundle of Rights*, *supra* n. 8 at 1256; Cf. Phillips, *supra* n. 11 at 1040; Cf. S.4, WIPO Model Provisions, 1978 *supra* n. 4.

93 Soderquist, *supra* n. 22.

94 Samuelson et al., *Manifesto, supra* n. 7 at 2380.

95 Stern, *Defining The Concept of Infringement*, *supra* n. 8 at 442.

96 Samuelson et al., *Manifesto, supra* n. 7 at 2398.

97 Stern, *Bundle of Rights*, *supra* n. 8 at 1252.

98 S.8, GNU GPLv3.

99 The MITI Report, *supra* n. 48 at 227 (Chapter V *Proposal for MITI Program Rights Act,* Outline(9)).

100 Stern, *Bundle of Rights*, *supra* n. 8 at 1257.

101 See, The MITI Report, *supra* n. 48 at 224.

102 Id at 227. (Chapter V *Proposal for MITI Program Rights Act,* Outline (10)(i)).

103 Samuelson et al., *Manifesto, supra* n. 7 at 2411.

104 Toeniskoetter, *supra* n. 24 at 79.

105 Id. at 80; Paley, *supra* n. 23 at 348; Stern, *Bundle of Rights*, *supra* n. 8 at 1247.

106 See, E.g. (U.S.) Peer To Patent <http://peertopatent.org/>; (Australia) Peer To Patent <http://www.peertopatent.org.au/>; (UK) Peer To Patent <http://www.ipo.gov.uk/peertopatent.htm>; (Japan) Peer To Patent <http://www.iip.or.jp/e/e_p2pj/> (last visited on November 7, 2011).