NOTE

SHARING THE INTEROPERABILITY BALL ON THE SOFTWARE PATENT PLAYGROUND

MICHAEL CHAPIN[*]

TABLE OF CONTENTS

[*] J.D. Candidate, Boston University School of Law, 2008; B.S. Materials Science Engineering, University of California, Berkeley, 1999.

I. INTRODUCTION

The debate over software patents has taken on many forms since the 1981 landmark Supreme Court decision in *Diamond v. Diehr* that opened up the door to patent protection for software.[1] After this decision, proponents of software patents pushed for broader protection of software and what it meant to meet the requirements of inventiveness, although not without significant growing pains. Through the 1980's and 1990's, the contentious debate over the patentability of software and computer programs largely revolved around whether software comprising mathematical algorithms should be appropriate subject matter for patent protection, whether patents on software provided over-protection and thus stifled competition, and whether these patents were even needed in addition to copyright protection in order to encourage innovation in the software industry.[2] Despite these valid and well articulated concerns against patent protection, the *In re Alappat* decision in 1994[3] and subsequent modifications to the United States Patent and Trademark Office ("USPTO") Examination Procedures made it significantly easier to achieve

[1] *See* Diamond v. Diehr, 450 U.S. 175, 184 (1981) (finding that an invention that contained a computer program used to control a rubber curing process was patentable subject matter). Although inventions involving computer programs had been issued patents by the patent office prior to *Diamond v. Diehr*, this case recognized for the first time that an invention involving a computer program satisfied the statutory subject matter requirements of the patent law, and could be patentable if it met the other patentability requirements of novelty, nonobviousness and utility. *Id.* at 191.

[2] *See, e.g.*, Simson L. Garfinkel, Richard M. Stallman & Mitchell Kapor, *Why Patents are Bad for Software*, ISSUES IN SCI. AND TECH., Fall 1991, at 52-55; Eloise Gratton, *Should Patent Protection be Considered for Computer-Software Related Inventions*, 7 COMP. L. REV. & TECH. J. 223, 241-52 (2003).

[3] *In re* Alappat, 33 F.3d 1526 (Fed. Cir. 1994).

patent protection for software.[4] Today, this debate is essentially moot as the ability to patent software is now deeply entrenched in both the U.S. patent system and within the international community.[5] However, unique aspects of the software industry, including incremental design,[6] complexity of software code,[7] and the tendency of "networking effects"[8] act as strong deterrents to new innovators unable to navigate this growingly complex and convoluted patent domain.[9] Additionally, software patents threaten the public's interest in the creation of standards and the ability of computer applications to interoperate and communicate effectively between each other or with operating systems.[10] In light of these concerns, the relevant debate related to software patentability has evolved into whether the rights afforded under a patent regime can be restricted in some way while still preserving the foundation of the patent system and its most fundamental goal of promoting the progress of science.[11]

Although the private sector and academia have considered limiting patent rights to solve the existing broad scope of software patentability,[12] the U.S. has

[4] *See* Examination Guidelines for Computer-Related Inventions, 61 Fed. Reg. 7478 (Mar. 29, 1996) [hereinafter 1996 Guidelines]. Obtaining a software patent was easier because software that merely modified a general purpose computer into a special purpose machine was now recognizable as statutory subject matter for patent protection. *Id*. at 7482-84.

[5] The international community, including Japan, Europe and most all industrialized nations have followed the U.S. in recognizing software as patentable subject matter to varying extents. *See generally* Jinseok Park, *Has Patentable Subject Matter been Expanded? – A Comparative Study on Software Patent Practices in the EPO, USPTO and the JPO*, 13 I.J.L. & I.T. 336 (2005) (detailing the evolution of patent protection for software in both Europe and Japan).

[6] *See* Julie E. Cohen & Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 CAL L. REV. 1, 41 (2001) (noting that "[i]t is rare for programs to be rewritten entirely from scratch; instead, innovation typically proceeds via a mix of new coding, modifications to some existing modules and subroutines, and either literal or functional reuse of others.").

[7] *See* David S. Evans & Anne Layne-Farrar, *Software Patents and Open Source: The Battle Over Intellectual Property Rights*, 9 VA. J.L. & TECH. 10, 20 (2004).

[8] Robert Hart, Peter Holmes, & John Reid, *The Economic Impact of Patentability of Computer Programs* (Intellectual Property Institute, London), Oct. 19, 2000, at 30 (describing the network effect as "[j]ust as the more users there are on a phone network the more valuable that network is to existing users, the more people use a piece of software the more existing users are likely to find themselves 'locked in'").

[9] *See* Evans & Layne-Farrar, *supra* note 7, at 22 (noting that "[a]s the tolls build during the technology's development path, later research could be discouraged altogether").

[10] *See* Jon T. Soma, Gus Winfield, & Letty Friesen, *Software Interoperability and Reverse Engineering*, 20 RUTGERS COMPUTER & TECH. L.J. 189, 194 (1994).

[11] *See* U.S. CONST., art. I, § 8, cl. 8.

[12] *See, e.g.,* Maureen A. O'Rourke, *Toward a Doctrine of Fair Use in Patent Law*, 100 COLUM. L. REV. 1177 (2000) (proposing a fair use exception for patent law); Jean Paul

done little in terms of Congressional action or legislative reform within the USPTO to address these growing concerns.[13] The U.S. government, through the USPTO, has always been a strong advocate for broad patent rights and has at times specifically opposed attempts to limit software patent protection.[14] However, in the early part of this decade, the European Community debated the scope of software patentability and addressed the need for software interoperability, a debate that still remains unresolved.[15] Additionally, Japan's Ministry of Economy, Trade and Industry ("METI") recently released a study on the potential detrimental impact of software patents on innovation,[16] and is

Smets & Hartmut Pilch, *Software Patentability with Compensatory Regulations: A Cost Evaluation*, UPGRADE, Dec. 2001, at 26-30, *available at* http://www.upgrade-cepis.org/issues/2001/6/up2-6Smets.pdf (proposing solutions such as a guarantee fund to protect small companies and open source developers from litigation, limiting patents rights either through interoperability exception or a limitations on the scope of claim structure, and a form of patent insurance); John S. Leibovitz, *Inventing a Nonexclusive Patent System*, 111 YALE L.J. 2251 (2002) (proposing a nonexclusive patent system where proof of independent creation would be a valid defense against infringement).

[13] The USPTO held hearings in 1994 that included commentary from industry representations on the topic of software-related inventions. *See* U.S.P.T.O *Public Hearing on Use of the Patent System to Protect Software-Related Inventions* (Feb. 10-11, 1994) (transcript available at http://www.uspto.gov/web/offices/com/hearings/software/arlington/vahrng.pdf). The USPTO most recently changed its examination guidelines related to software patents in 1996. *See* 1996 Guidelines, *supra* note 4. The FTC also recently conducted a study that investigated among other things, the effect of the patent system on the software industry. *See* FEDERAL TRADE COMMISSION, TO PROMOTE INNOVATION: THE PROPER BALANCE OF COMPETITION AND PATENT LAW AND POLICY ch. 3, at 44-56 [hereinafter FTC REPORT]. However, this study has not resulted in any changes to the patent system specifically aimed at software inventions or interoperability.

[14] *See* Letter from USPTO, U.S. Comments on the Draft European Parliament Amendments Regarding the Proposed European Union Directive on the Patentability of Computer-Implemented Inventions (Sep. 16, 2003), *available at* http://www.aplf.org/mailer/USCommentsPatentCompImplInv.pdf [hereinafter USPTO Letter to European Commission]. This letter was sent in 2003 as opposition to the European Union's consideration of the European Parliaments proposed amendments that would have limited the scope of software patent protection in Europe.

[15] *See generally* Commission Proposal for a Directive of the European Parliament and of the Council on the Patentability of Computer-Implemented Inventions, COM (2002) 92 final (Feb. 20, 2002) [hereinafter CII Directive]. This debate largely focused on the technical character that should be required of a software invention for patentability and whether there should be some limit on the scope of enforceability of software patents against infringing uses needed to achieve interoperability. For a brief discussion, *see* Park, *supra* note 5, at 348-50.

[16] *See* METI COMMERCE AND INFORMATION POLICY BUREAU, INTERIM REPORT OF "STUDY GROUP ON THE LEGAL PROTECTION OF SOFTWARE AND PROMOTION OF INNOVATION" (Oct. 11, 2005), *available at* http://www.meti.go.jp/english/information/downloadfiles/051017LegalProtectionSoftware.pdf [hereinafter METI Interim Report]. The Japanese Patent Office

currently considering changes to its "Rules on Electronic Commerce" that would limit the scope of rights for software patents by specifically adding exceptions for interoperability.[17] The private industry, both internationally and within the U.S., has also tried to address interoperability concerns through standard setting organizations and the Open Source Movement.[18] With the recent Supreme Court decision *eBay v. MercExchange* recognizing some limitation on patent rights,[19] and with the U.S. currently looking at other aspects of our patent system as they relate to harmonizing our system with the international community,[20] the U.S. needs to revisit the current scope of rights afforded to software patents.

Solutions to this increasing need for software interoperability should address both the ability to legally access the underlying source code forming the basis of a patent and the ability to use that patented source code to achieve interoperable products without fear of patent infringement. Ways of better facilitating accessibility such as through increased patent disclosure requirements, especially in the face of growing protective security measures and encryption for software, are largely beyond the scope of this Note. Focusing on just the patent protection aspect of this problem, Part II of this Note will first look at the history of intellectual property protection for software while Part III will look at how software interoperability is threatened by a system of patent protection in more detail. Part IV will then examine proposed reforms by both the international community and the private sector aimed at ensuring and protecting interoperability as well as looking at the recent *eBay v. MercExchange* Supreme Court decision and its potential implications for interoperability. Part V will propose legislative complements to private sector solutions to better protect interoperability, examine the proposed international reforms and whether they can be applied to or harmonized with the U.S. patent system, and whether recent U.S. court decisions coupled with legislative amendments can provide the needed protection for software interoperability within the U.S. patent system.

(JPO) is an external bureau of the Ministry of Economy Trade and Industry.

[17] *See* Press Release, METI, Request for Public Comments on Rules on Software Intellectual Property Rights (June 13, 2006), *available at* http://www.meti.go.jp/english/newtopics/data/n060613e.html. The proposed rule changes can be found in Japanese at http://www.meti.go.jp/press/20060613001/press.pdf.

[18] These organizations include the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO) among others. For a discussion of the Open Source movement and standard setting, see generally Evans & Layne-Farrar, *supra* note 7, at 3-5, 15-18.

[19] eBay v. MercExchange, L.L.C., 547 U.S. 388 (2006).

[20] *See* Patent Reform Act of 2007, H.R. 1908, 110th Cong. (2007). The Patent Act is intended to harmonize certain aspects of the U.S. Patent system with practices in the internationally community, most notably adopting the first-to-file practice of determining priority of inventorship. *Id.* § 3.

II. BACKGROUND ON SOFTWARE INTELLECTUAL PROPERTY

A. *Trade Secret and Copyright Protection of Software in the U.S.*

When software began to be incorporated into commercial products in the 1960's and 1970's, trade secret law was the principal form of legal protection that a company used to protect its software from being copied or misappropriated.[21] When the software industry was in its infancy, the need for interoperability among computer software was not very important because products were either task specific to customer needs or part of large expensive mainframe machines that did not require the need to communicate with competing software.[22] Although copyright was granted to software during this time,[23] certain limitations made copyright protection underutilized until the Copyright Act was revised and amended in 1976.[24]

Prior to this amendment, Congress formed the National Commission on New Technological Uses of Copyrighted Works ("CONTU") to make recommendations for changes to the Copyright Act.[25] Although software was not mentioned in the amended act of 1976, CONTU's recommendations to include computer programs in the form of source or object code as literary works were largely adopted by Congress in 1980 and incorporated into the Copyright Act.[26] With these amendments, specifically the protection of computer programs in object code and source code form as literary works, copyright became widely used as a means to protect software from unauthorized copying. However, although the use of software had become fairly widespread and interoperability concerns were known at the time of these amendments, CONTU "failed to address the key interoperability challenges that were beginning to confront the emerging mass-market software industry."[27]

[21] *See* Bradford L. Smith & Susan O. Mann, *Innovation and Intellectual Property Protection in the Software Industry: An Emerging Role for Patents?*, 71 U. CHI. L. REV. 241, 243 (2004).

[22] *Id.*

[23] *Id.* at 244 (observing that 1,205 copyrights were granted for software between 1964 and 1977).

[24] *See id.* at 245-247 (these limitations largely revolved around copyright protection for computer software in object code form). Software was given interim protection through Section 117 of the 1976 Amendments while awaiting CONTU's final report.

[25] *See generally* Robert P. Merges, *One Hundred Years of Solicitude: Intellectual Property Law: 1900-2000*, 88 CAL. L. REV. 2187, 2198-2200 (2000).

[26] *See* Jeffrey A. Andrews, Comment, *Reversing Copyright Misuse: Enforcing Contractual Prohibitions of Software Reverse Engineering*, 41 HOUS. L. REV. 975, 981 (2004-2005) (citing Act of Dec. 12, 1980, Pub. L. No. 96-517, 94 Stat. 3015 (1980)).

[27] Smith & Mann, *supra* note 21, at 247 (citing NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, Final Report, at 34 (1979)). Although

The problem that copyright protection raised regarding interoperability was not that the functional aspects of the computer code would be protected under copyright, but rather that the reverse engineering of the software to gain access to these unprotected functional components of the code would infringe on the expressive elements that were protected under copyright.[28] U.S. courts resolved this debate by recognizing a fair use defense to decompiling and reverse engineering software "solely in order to discover the functional requirements for compatibility."[29] Access to a program's copyrighted source code was essential to ensure interoperability since it would reveal the program's non-copyrightable interface specifications needed to effectively share data between two programs.[30] This outcome was largely a result of courts recognizing the growing concern with software that copyright owners could otherwise force consumers to continue using a particular standard or interface if these unprotected elements could not be accessed.[31]

As software programs became more complicated and skill in computer programming grew more commonplace, the bulk of the value in software code began to reside in the ideas behind the programs rather than the time spent on writing the source code and in its expression.[32] Copyright protection of the software code was limited by independent creation and by this inability to protect the idea behind the software code as a result of the idea-expression dichotomy doctrine.[33] This need for additional protection drove large software

CONTU seemed to recognize that interoperability concerns might conflict with copyright protection, the commission did not account for this in its proposed changes to the Copyright Act. *See id.*

[28] For a discussion on reverse engineering and software decompilation, *see* Gerald Dworkin, *The Concept of Reverse Engineering in Intellectual Property Law and its Application to Computer Programs,* 1 AUSTL. INTELL. PROP. L.J. 164, 164-173 (1990).

[29] Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1522 (9th Cir. 1993) (recognizing a fair use exception to decompile software in order to achieve interoperability). *See also* Sony Computer Entertainment, Inc. v. Connectix Corp., 203 F.3d 596, 602-04 (9th Cir. 2000) (recognizing intermediate copying in the form of reverse engineering to access functional elements of a software program to achieve interoperability). Reverse engineering to achieve interoperability has since been codified under the Digital Millennium Copyright Act of 2000. *See* 17 U.S.C. § 1201(f) (1998).

[30] *See* Smith & Mann, *supra* note 21, at 249.

[31] *See* Lotus Dev. Corp. v. Borland Intl., Inc., 49 F.3d 807, 821 (1st Cir. 1995) (recognizing the public's interest in allowing interface aspects of the Lotus interface to be copied and thus limiting the extent of the copyright protection as opposed to enforcing the copyright and essentially forcing consumers to "remain captives of Lotus because of an investment in learning made by the users and not by Lotus").

[32] *See* Bitlaw Home Page, *Why Protect Software through Patents*, http://www.bitlaw.com/ software-patent/why-patent.html (last visited Mar. 24, 2008).

[33] A well-known example of bypassing copyright protection through independent creation is Phoenix Software Associate's independent creation of IBM's PC compatible

companies to seek stronger protection for their software. The *Diamond v. Diehr* decision in 1981 and subsequent U.S. court decisions recognizing software patents bore the fruits of these efforts.

B. Patent Protection of Software in the U.S.

As copyright law failed to provide complete protection, stronger protection in the form of patents was sought to provide the necessary economic incentives to continue to spur creativity and address the growing concerns that copyright under-protected software.[34] The scope of patent protection grew broader as the requirements of acceptable software claims were relaxed with subsequent decisions like *In re Alappat* in 1994 that allowed claims to include software running on a general purpose computer.[35] Ultimately, this led to the amendment to the USPTO Examination Guidelines in 1996 that gave broad patent protection to software innovation,[36] thus threatening the inherent balance between software protection and the recognized public interest in software interoperability that the courts had already struck in the copyright context.[37]

Early adaptation of software to the patent system and issuing patents for either obvious or non-novel software seemed to raise potentially more serious issues than the threat to interoperability.[38] In fact, patent protection was probably sought as a means to protect the advantages gained from the "networking effect" thus enabling dominant positions to be formed by leveraging these patents against any future competitor.[39] As a result, much of the early debate over software patents focused on how to reduce the granting of "bad" software patents – software inventions unworthy of patent protection.[40]

ROM BIOS in the 1980's through utilization of a clean room technique that isolated the engineers who were decompiling the code from the engineer who was creating the cloned BIOS. For a general discussion about this technique and Phoenix's success, see Russell Moy, *A Case Against Software Patents*, 17 SANTA CLARA COMP. & HIGH TECH. L.J. 67, 70-73 (2000).

[34] *See* Leibovitz, *supra* note 12, at 2284-85.

[35] *In re* Alappat 33 F.3d 1526, 1544-45 (Fed. Cir. 1994).

[36] *See* 1996 Guidelines, *supra* note 4. By allowing functional claiming in the form of a means or step for performing a specific function, compatible products became effectively barred. *See* Moy, *supra* note 33, at 90 ("functional claiming. . .grants monopoly rights to the patent holder for the element that is functionally claimed, barring any practical competition").

[37] *See e.g.,* Sega Enters., 977 F.2d 1510, 1522 (9th Cir. 1993).

[38] *See* Cohen & Lemley, *supra* note 6, at 12-14.

[39] *See* Tord Jansson, *Why Software Shouldn't be Covered by Patents*, http://bladeenc.mp3.no/articles/software_patents.html (last visited Mar. 24, 2008).

[40] *See, e.g.*, Pamela Samuelson, *Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 EMORY L.J. 1025, 1138-39 (1990).

This emphasis was partially caused by the USPTO's bias toward hardware over software and its reluctance to hire qualified examiners with software backgrounds.[41] Granting patents for software that lacked novelty or non-obviousness was partly caused by the fact that prior art for software was found "outside the areas in which the PTO has traditionally looked – previously issued patents and previous scholarly publications."[42] Recent USPTO rule changes attempt to resolve this by accepting public submissions of prior art to patent examiners,[43] but this still rarely occurs.[44] The USPTO also partially addressed concerns of inadvertent infringement when it began publishing patent applications, but the sheer mass of patents and patent applications to search through coupled with the lack of source code in the description make discovering potentially infringing patents a daunting task for any software inventor.[45]

Although the adoption of the patent system to software in the U.S. has not been free of difficulties and growing pains as detailed above, the system has improved significantly. The alternative system, in which there is no patent protection for software, could seriously hinder small companies from receiving funding to spur innovation.[46] Ultimately, a reversion back to a system of trademark and copyright would likely lead to less disclosure of inventive code

[41] Since pure software patents were not issued in the 1980's, and since the software aspects of an innovation had to be linked to a physical apparatus or a process in order to meet statutory subject matter requirements, the USPTO did not start hiring adequately trained examiners with programming backgrounds until the mid 1990's. *See* Evans & Layne-Farrar, *supra* note 7, at 14.

[42] John R. Allison, *The Business Method Patent Myth*, 18 BERKELEY TECH. L.J. 987, 1013 (2003) (noting that developments in the computer software industry are not commonly documented in scholarly publications but rather discussed only in text books or user manuals that will typically not be found by patent examiners).

[43] See 37 C.F.R. § 1.99 (2005).

[44] This lack of third party submissions can be attributed to the fact that third parties are often unaware of patent applications being published that are of interest to them, to the fact that third parties may be unaware that they can even submit prior art, and because current law prevents commentary from being submitted along with the prior art. *See* Manny W. Schecter, *Open Collaboration is Medicine for Our Ailing Patent System*, BNA'S PAT., TRADEMARK & COPYRIGHT J., Oct. 20, 2006, at 684, *available at* http://dotank.nyls.edu/communitypatent/BNA_10-20-06.html. The inability to submit commentary deters third party submissions because of the fear that examiners may not appreciate the value of the prior art and thus make it more difficult to overcome a presumption of validity at a later date if a reference has been placed on the record. *See id.*

[45] *See* Leibovitz, *supra* note 12, at 2285.

[46] John R. Allison, Abe Dunn & Ronald J. Mann, *Software Patents, Incumbents and Entry*, FRONTIERS OF IP, Nov. 11, 2006, at 41, *available at* http://www.utexas.edu/law/conferences/ip/MannPaper.pdf (finding that empirical evidence shows a strong correlation between acquiring software patents and a startup firms progression through rounds of venture-capital funding).

and thus decreased accessibility to achieve interoperability.[47] However, neither the USPTO nor Congress has yet to propose a solution to address software interoperability needs, a concern that the international community has attempted to address while following the U.S. in the acceptance of software patentability.

C. International Protection of Software (Europe and Japan)

1. Software and Intellectual Property in Europe

The international community for the most part has followed America's lead in recognizing both copyright and patent protection for software to a varying extent within their respective patent systems. The European Union recognizes copyright protection for software, and has specifically codified an interoperability exception similar to the copyright fair use case law and statutory exception in the U.S.[48] The E.U. provision states that "[a]uthorization of the right holder shall not be required where reproduction of the code. . . [is] indispensable to obtain the information necessary to achieve the interoperability of an independently created computer program with other programs."[49] With regard to software patentability, although computer programs "as such" are barred from patentability under the European Patent Convention of 1973 ("EPC"),[50] the European Patent Office ("EPO")[51] has narrowly interpreted the language that specifically excludes computer programs "as such" in order to allow for the patentability of inventions containing computer programs.[52] The EPO's Examination Guidelines now specifically address computer implemented inventions[53] and the proper

[47] *See* Evans & Layne-Farrar, *supra* note 7, at 26. Specifically with trademark and to a lesser extent copyright, interoperability would be threatened unless adequate protective mechanisms preventing access to the software could be developed.

[48] *See* 17 U.S.C. § 1201(f) (1998).

[49] Council Directive 91/250, art. 6, 1991 O.J. (L 122) 42 (EC).

[50] *See* Convention on the Grant of European Patents, art. 52, October 5, 1973, 13 I.L.M. 268 (entered into force July 10, 1977). Article 52 states: "The following in particular shall not be regarded as inventions within the meaning of paragraph 1. . .(c) schemes, rules and methods for performing mental acts, playing games or doing business, and programs for computers; (3) The provision of paragraph 2 shall exclude patentability of the subject-matter of activities referred to in that provision only to the extent to which a European patent application or European patent relates to such subject-matter or activities as such." *Id.* at art. 52.

[51] The EPO is the administrator of the EPC.

[52] For an analysis of the evolution of EPO's case law leading to this current interpretation of the EPC, *see* Park, *supra* note 5, at 337-42.

[53] The EPO defines a computer-implemented invention as "an invention that works by using a computer, a computer network or other programmable apparatus. To qualify, the invention also needs to have one or more features which are realized wholly or partly by

structure that claims must meet in order to satisfy the statutory subject matter and technical character requirements.[54]

Under this interpretation by the EPC, thousands of software patents have been granted by the EPO;[55] however, there still remains significant legal uncertainty as to the enforceability of these patents because an EPO patent can still be invalidated by an individual country within the European Union.[56] As a result, very few software patents have been litigated within Europe because of the uncertainty and fear of being invalidated by a national court.[57] Thus, although Europe has seemingly embraced software patentability, this uncertainty of patent protection for software is the subject of significant debate with the European community. However, unlike the U.S., interoperability concerns are at the heart of this debate and the proposed solutions, as Europe seeks to find a solution for software patents that all can agree on.[58]

2. Software and Intellectual Property in Japan

Following the lead of the U.S., Japan also expanded the scope of its copyright and patent laws to include software. In several instances, Japanese courts had recognized computer programs as protectable under the copyright law prior to a statutory amendment in the middle of the 1980's.[59] At that time,

means of a computer program." EPO Home Page, Computer-Implemented Inventions (CII), http://www.epo.org/topics/issues/computer-implemented-inventions.html (last visted Mar. 24, 2008). The EPO's Examination Guidelines on computer implemented inventions were most recently updated in 2005. *See Guidelines for Examination in the EPO*, pt. C, ch. IV, § 2.3.6 (2005), *available at* http://www.epo.org/patents/law/legal-texts/html/guiex/e/c_iv_2_3_6.htm.

[54] *See In re* Vicom Sys., Inc., T 0208/84, 1986 O.J.E.P.O 14, 19 (Tech. Bd. App. 1986).

[55] *See* EUROPEAN PATENT OFFICE 1994 ANNUAL REPORT, PATENTING COMPUTER SOFTWARE (1995) (noting over 11,000 software-invention patents being granted by the EPO as of 1995).

[56] *See, e.g.*, CFPH LLC v. Comptroller-General of Patents, Designs and Trade Marks, 2005 EWHC 1589 (Pat), §§ 54-56 (July 21, 2005) (U.K.) (recognizing that a uniform interpretation of Article 52 of the EPC is desirable, the U.K. High Court reaffirmed that the EPO has no jurisdiction over patent law in the U.K., and that its decisions are persuasive, not prescriptive). The ability of an individual country to overrule an EPO patent is real risk since each country that is a signatory under the EPC is capable of interpreting this apparently vague "as such" language differently.

[57] *See* Smets & Pilch, *supra* note 12, at 26 (observing that the risk of an EPO patent being voided by a national court creates a form of self-regulation and prevents abuse by software patent holders).

[58] *See* discussion *infra* Part IV.A.1.

[59] *See* Rohan Mishra, Comment, *Reverse Engineering in Japan and the Global Trend Towards Interoperability*, 4 MURDOCH U. ELEC. J.L. 1, 4-6 (1997) (citing Taito K.K. v. K.K. ING Enterprises, 1129 HANREI JIHO 120 (Tokyo D. Ct., Sep. 28, 1984) (holding that although computer programs were not listed as protectable works under the copyright law,

a sui generis form of protection that would have limited the duration of protection and incorporated a form of compulsory license was seriously considered, but as a result of strong international opposition by both the U.S. and Europe, the amended Japanese copyright laws included computer programs as protectable subject matter with no specific limitations.[60] Although limited Japanese case law has addressed some rights to decompile and reverse engineer software that could be interpreted to allow for an interoperability exception, proposed amendments to Japan's Copyright Law have never been implemented.[61]

Japan also largely followed the U.S. in recognizing computer programs as statutory subject matter for patent protection. The evolution of software patentability in Japan was not prompted by case law, but rather through the Japanese Patent Office ("JPO") interpretations and application of the Japan's Patent Act through revisions of the JPO's Examination Guidelines.[62] These current guidelines define what software-related inventions constitute statutory subject matter,[63] and the Japanese legislature amended the Patent Law in 2002 to allow patentability of a computer program claim itself by treating it as a tangible entity.[64]

Although Japan has implemented a broad scope of patent protection for software, a recent study by Japan's Ministry of Economy, Trade and Industry ("METI") suggests that Japan may be considering backpedaling from its

that the work was a creative expression and thus a protectable work of authorship)).

[60] *See id.* at 7-9.

[61] *See id.* at 21 (interpreting Japanese case law that restricts decompilation where use creates a program that is substantially similar should allow for decompilation "for the purpose of achieving interoperability, as long as the final product is not substantially similar to the original").

[62] The JPO first modified its examination procedures in 1993 that specifically stated an invention involving software was not excluded from patentability. In 1997, the JPO included storage media containing programs as patentable subject matter and again in 2000 to treat a software program itself as tangible and thus patentable. *See* Park, *supra* note 5, at 365-66. An invention must still meet the requirements of a highly advanced creation of technical ideas by which a law of nature is utilized. *See* Patent Act, Law No. 121, April 13, 1959, art. 29(1)-(2) (amended Dec. 22, 1999) (Japan).

[63] JPO's Examination Guidelines for Patent and Utility Model, pt. VII, ch. 1, § 2.2.1(1) (2000), *available at* http://www.jpo.go.jp/tetuzuki_e/t_tokkyo_e/1312-002_e.htm (stating that a software program is patentable if "[a]s a result of reading the software into the computer, the information processing equipment (machine) or operational method. . .is constructed by concrete means in which software and hardware resources are cooperatively working so as to realize arithmetic operation or manipulation of information. . .").

[64] *See* Toru Yamauchi, *Legislative Changes in Japan and Their Effect on Software Patents*, THOMSON SCIENTIFIC, August 2002, *available at* http://scientific.thomson.com/free/ipmatters/sbm/8180030 (noting that the new law clears up ambiguity as to whether software for patent purposes could be an entity in itself or required to be in the form of a method).

current course of providing expansive protection for software patents.[65] The Japanese study based a large part of its reasoning on growing concerns as a result of certain unique characteristics of the software industry detailed in a 2003 report by the U.S. Federal Trade Commission.[66] These characteristics include cumulative innovation, low capital costs, rapid rate of technological change, alternative means besides patents for fostering innovation, and the infancy of patent protection in the software industry.[67] Part III will further highlight these concerns, while Part IV will look at the different international solutions aimed at ensuring software interoperability, including this METI study and the resulting proposed rule changes that would create an interoperability exception for software patents under Japan's patent system.

III. PROBLEMS WITH THE CURRENT U.S. PATENT SYSTEM RELATED TO SOFTWARE INTEROPERABILITY

The software industry and computer programs in general are particularly unique compared with other technologies.[68] The software industry is characterized by the incremental nature of programming where software is often built upon components of pre-existing programs.[69] There is a strong need for programs to interoperate and to ensure compatibility between file formats, network protocols and interfaces, as well as the need for common languages and standards.[70] However, the current patent system substantially constrains software improvements and innovation. This is amplified by a characteristic common in the software industry known as "network effects," a phenomena that locks in users of that technology as that particular software becomes dominant in the market.[71] Although an alternative to an interface or

[65] *See* METI Interim Report, *supra* note 16.

[66] *See generally* FTC Report, *supra* note 13.

[67] *Id.* at 55-56.

[68] This uniqueness raises concerns when software inventions are applied to a patent system. *Id.*

[69] *See id.* at ch. 3, 44-45. This incremental effect is naturally amplified as the software industry continues to evolve and computer programs increase in complexity.

[70] *See* Smets & Pilch, *supra* note 12, at 26. The growing complexity of the software industry also creates an ever increasing need for software to be able to communicate with each other and for the need of established standards and common languages. *See* Jason V. Morgan, Open Source Software and Software Patents: Finding the Common Ground in a Patent Pool 3 (2002) (unpublished B.S. thesis, University of Utah), *available at* http://opensource.mit.edu/papers/morgan.pdf.

[71] *See* Hart, Holmes, & Reid, *supra* note 8, at 30 (analogizing the network effect to a telephone system prior to recent technology, where users had no real other choice but to use the telephone network to communicate with others since everyone else was on the network). This phenomena locks in consumers both by making belonging to the network more valuable since a large number of other consumers use the network as well as through switching costs, where knowledge and skill will have to be relearned to some extent if the

standard is often achievable, a user base that is invested in that standard or interface entrenches that technology, creating significant hurdles to a future entrant.[72] This phenomena essentially creates a *de facto* standard that "arise[s] from the operation of the market, as consumers gravitate towards a single product or protocol and reject its competitors."[73] Where a technology that becomes a *de facto* standard is controlled by a single patent holder, he has significant power and control over every company that seeks to create a compatible software product.[74] Unlike other industries where a certain technology can be designed around through innovation and new inventions, the ability to achieve a design around is virtually impossible in the software industry while still achieving interoperability, thus creating a significant threat to the public's interest in competition and compatible software.[75]

Although this threat is real, it is often in a company's best interest to license its technology because this creates additional applications that help promote that technology into a standard as it becomes more commonplace. "Market forces work to ensure interoperability without government intervention because a software developer would not enjoy commercial success unless it enabled, facilitated, and indeed promoted interoperability."[76] However, these market forces cannot be relied upon to promote interoperability where a potential innovator cannot afford a license, where a potential licensor is a large

user switches software. *See* Elliot Maxwell, *Open Standards, Open Source, and Open Innovation – Harnessing the Benefits of Openness*, INNOVATIONS: TECHNOLOGY, GOVERNANCE, GLOBALIZATION, Summer 2006, at 126 n.15 (citing Metcalfe's law for valuing a network as being proportional to the square of the number of people already owning or using that product).

[72] *See* Grant C Yang, *The Continuing Debate of Software Patents and the Open Source Movement*, 13 TEX. INTELL. PROP. L.J. 171, 186 (2005).

[73] *See* Mark A. Lemley, *Intellectual Property Rights and Standard-Setting Organizations*, 90 CAL. L. REV. 1889, 1899 (2002). *See also* Morgan, *supra* note 70, at 3 (using as an example the need for a new software program to interface with a Microsoft Word's file name, but if the format of the file name is arbitrarily patented, that the new software would be incompatible with the de facto standard set by Microsoft unless it either received a license or infringed the patent). The ability to patent this first to market phenomena that often leads to a *de facto* standard can give the original developer immense leverage and control over competing products.

[74] *See* Evans & Layne-Farrar, *supra* note 7, at 16.

[75] *See* Moy, *supra* note 33, at 94 (noting that "similar but functionally different programs would not enjoy many of the features that compatible software would offer customers. . . compatible software would allow customers to avoid the waste associated with learning to use new software packages, and it would provide customers with more software options. . . [and] may also facilitate the creation of networks and file sharing.").

[76] S*ee* Letter from The Computing Technology Industry Association, Response to Japan's Ministry of Commerce and Industry (METI) Rules Concerning Software Related Intellectual Property (Jul. 12, 2006), *available at* http://comptia.org/issues/docs/SRIP%20JAPAN%20V1%203_final%20_3_.pdf [hereinafter *CompTIA Response*].

competitor of the *de facto* standard holder, or where there is a small economic incentive to license the technology.[77] Of course, a competing company could attempt to migrate to a completely different standard, but this would significantly destabilize the standardization process.[78] As an example of the importance of being able to interoperate freely within the software industry, the key success to the Internet is largely attributed to "interoperability between hardware and software and the cross-network interoperability of software."[79]

Recognizing the public's interest in software interoperability, the private sector has implemented partial solutions such as standard setting bodies, cross license agreements and open source. However, as this Note will discuss, some degree of government legislation similar to that in the copyright context is necessary as a complement to these private solutions. Contrary to the U.S. however, Europe and Japan have already attempted such legislation that would take into account this need for software interoperability within a patent system.

IV. PROPOSED SOLUTIONS BY THE INTERNATIONAL COMMUNITY AND PRIVATE SECTOR AND THE IMPLICATIONS OF RECENT U.S. CASE LAW

A. *Proposed Solutions Addressing Software Interoperability by the International Community*

1. European CII Directive

The need to ensure interoperability between software applications within a patent system has already been recognized as a growing concern by governments within the international community. With significant ambiguity surrounding software patents in Europe largely as a result of the differing interpretations of the "as such" language under the EPC,[80] the European Union

[77] *See* Smets & Pilch, *supra* note 12, at 26 (citing Apple's exclusive license on Sorenson patents covering digital video compression and its refusal to sublicense the technology to competitors as an example where competing products like Real Player and Windows Media Player were prevented from interoperating and playing files compressed by this technology).

[78] *See* 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, 287 (2002) (observing that "[i]f this scenario were taken to its extreme limits, this would lead to an ever-increasing balkanization of computers"). Under this scenario, "[s]oftware programmers would have to engineer an entire suite of programs and platforms for any one user, because there would be no guarantee that the user could integrate another program or platform." *Id.*

[79] PBT CONSULTANTS LTD., THE RESULTS OF THE EUROPEAN COMMISSION CONSULTATION EXERCISE ON THE PATENTABILITY OF COMPUTER IMPLEMENTED INVENTIONS 30 (2000), *available at* http://ec.europa.eu/internal_market/indprop/docs/comp/softanalyse_en.pdf.

[80] *See* Convention on the Grant of European Patents, art. 52, Oct. 5, 1973, 13 I.L.M. 268 (entered into force July 10, 1977).

revisited the software patent debate in 2002 with a proposal on the patentability of computer-implemented inventions ("CII Directive").[81] The original proposal by the European Commission recognized the need for interoperability and contained limited language related to decompilation in the form of reverse engineering to achieve interoperability for patented software.[82] However, this language only referred to decompilation, and did not make clear whether a patent could subsequently be used to exert a claim of infringement against interoperating products.[83] Upon review by the European Parliament, a clear interoperability exception similar to that afforded under copyright was added to the draft proposal. The exception stated "wherever a use of patented technique is needed for a significant purpose such as ensuring conversion of the conventions used in two different computer systems or networks so as to allow communication and exchange of data content between them, such use is not considered to be a patent infringement."[84] Under this language, an interoperability exception would not render a software patent useless; rather, it would merely ensure the ability of new software to communicate freely with the patented software or to allow for new standards to be created that could communicate with and convert output from previously patented standards.[85] However, the proposed changes were subsequently rejected by the European Commission as a result of heavy lobbying from pro-patent groups and the U.S. government,[86] and the CII Directive was ultimately defeated in a second

[81] *See* CII Directive, *supra* note 15.

[82] *Id.* at art. 6.

[83] *See* Mikko Valimaki, *Software Interoperability and Intellectual Property Policy in Europe*, 3 EUR. REV. OF POL. TECH., Dec. 2005, at 5 n.5, *available at* http://www.politech-institute.org/review/articles/VALIMAKI_Mikko_volume_3.pdf.

[84] *See* EUR. PARL. DOC. (A5-0238/2003) 16 (2003).

[85] *See* FFII Position Paper, *Interoperability and the Software Patents Directive: What Degree of Exception is Needed* (Nov. 16, 2003), *available at* http://eupat.ffii.org/papers/eubsa-swpat0202/itop/eubsa-itop.en.pdf. An interoperability exception was supported by some software firms, most notably SUN Microsystems, who issued a position paper on the topic. Sun emphasized that patent protection "should be balanced to allow for the creation of products which can interoperate with the protected products to safeguard competition in the sector and to provide greater choice and lower costs for consumers." Sun Microsystems Home Page, Position Papers: Software Patents, http://web.archive.org/web/20061218011518/http://www.sun.com/aboutsun/policy/software_patents.html (last visited Mar. 24, 2008). However, Sun only supported a more narrow exception that would permit uses that were "indispensable" in order to create a product that could interoperate with products covered under a patented technology. *Id.*

[86] The proposed amendment for interoperability was strongly opposed by the Business Software Alliance (BSA), consisting of Microsoft, Apple, and other U.S. software firms. The USPTO also expressed strong concerns with the broad exception the language might afford would-be–infringers, and that the exception would essentially allow infringement without a showing or determination of anti-competitive activity. *See* USPTO Letter to

reading by the European Parliament in July 2005 as a result of the proposed changes from its first reading being rejected.[87] Thus the status of software patents in general, and the status of interoperability between patented computer programs and application software inventions, remain in a state of uncertainty within Europe.

2. Japanese Proposed Rule Changes

The Japan's Ministry of Economy, Trade and Industry ("METI") also recently began to seriously address the issue of interoperability in the software sector with the release of a study on the legal protection of software in 2005.[88] Although the study recognizes that the granting of patent protection plays a strong role in the promotion of innovation in the software sector, it also notes that software inventions are unique based on software's inextricably layered and linked structure, and that the patent system in Japan does not currently take these characteristics into account to effectively promote and foster continuous innovation.[89] Characterizing software by its ability to function only through the communication with other software, METI argues that a competitive environment requires the ability for upper-level software to use functions and rules of lower level software without restriction.[90] The study also recognizes the "network effect" caused by software that accounts for a major portion of a market, especially as it relates to interoperability and interfaces, observing that "factors such as economies of scale and high cost of migration to another platform tend to result in a long-lasting monopolistic market and generates adverse effects on innovation because of inhibited competition."[91]

To promote innovation and to address the needs of interoperability, the study proposes three possible changes to the patent law specifically related to software. These changes include a restriction on the exercise of patent rights on a case-by-case basis by using some type of abuse of rights principle, a compulsory license granted on a case-by-case basis, or a general restriction on the exercise of patent rights through an amendment to the Patent Act specifically aimed at limiting the effects of software patents.[92] METI lists several conditions that would constitute patent abuse under its first proposal – tying practices by requiring a license for another patent, mandating that related patents acquired after licensing to be assigned over, and merely acting to

European Commission, *supra* note 14, at 3-4.

[87] *See* Ingrid Marson, *Software patent directive rejected*, ZDNET UK, July 6, 2005, http://news.zdnet.co.uk/itmanagement/0,1000000308,39207478,00.htm.

[88] *See generally* METI INTERIM REPORT, *supra* note 16.

[89] *See id.* at 1.

[90] *See id.*

[91] *Id.* at 2. This effect is even more pronounced where the patented technology has been standardized.

[92] *Id.* at 3-4.

impede interoperability.[93]

METI has since followed up on this study by proposing new rules related to software patentability that would adopt a form of an abuse of rights principle proposed in the interim report.[94] The new rules would allow a finding of an abuse of right in situations where:

> bad faith is found in the manner of enforcing the right, such as the intention to unreasonably cause a disadvantage to parties affected by such enforcement of the right . . . [or] [w]here an extremely large disadvantage in contrast to the benefit to be gained by enforcing such right is caused to parties affected by such enforcement of the right as well as to society.[95]

These proposed rule changes are intended to "clarify the scope of abuse of rights applicable where exercising software patent rights hinders promotion of software innovation, such as ensuring software interoperability."[96] The current proposal, if implemented, would seem to allow for infringing uses without a license agreement where such use is needed to achieve interoperability. This would vary significantly from recent changes to the patent misuse doctrine as implemented in the U.S. that requires a showing of anticompetitive effect on the part of the patent holder in an antitrust case based on product tying.[97] METI's proposed rules would provide for a complete release of liability for infringement where presumably there can be shown a need for interoperability.[98] METI is currently seeking public comments and has naturally received opposition from several software industry groups, including the Business Software Alliance and the Computer Technology Industry Association.

3. Flaws with the European and Japanese Solutions

Both the European and Japanese proposals to codify a specific exception for

[93] *Id.* at 4.

[94] *See* Press Release, METI, Proposed Rules concerning Software-Related Intellectual Property (June 13, 2006), *available at* http://www.meti.go.jp/press/20060613001/press.pdf [hereinafter *METI Proposed Rules*].

[95] CompTIA Response, *supra* note 76, at 3 (citing METI Proposed Rules).

[96] Press Release, METI, *supra* note 17.

[97] The U.S. Supreme Court also recently clarified the ambiguity as to whether a patent covering a tying product creates a presumption of market power under antitrust laws. *See Illinois Tool Works Inc. v. Independent Ink, Inc.*, 547 U.S. 28, 44-45 (2006) (holding that the mere fact that a tying product is patented does not support such a presumption). This recent case was aimed at harmonizing the antitrust laws with the Patent Act that had already eliminated this market power presumption in patent misuse cases. *See id.*

[98] *See* Letter from Seow Hiong Goh, Director of Software Policy, Asia, Business Software Alliance, to METI, Comments on the Draft of Rule Concerning Software Related Intellectual Property 4-5 (July 12, 2006), *available at* http://www.bsa.or.jp/file/BSA_Comments_English_060712.pdf.

interoperability into their patent systems would significantly improve the ability of software to communicate freely between application programs and lower level functions and operating systems. However, an interoperability exception should be narrowly defined so as to not add to the already existing confusion regarding potential infringement of software patents.[99] Software firms that supported the CII Directive's attempt to incorporate an interoperability exception proposed modified language such as "indispensable" in order to better define the scope of the exception.[100] The USPTO in its letter to the European Commission opposed the language in the CII directive by arguing that the proposed interoperability language allowed for too broad of infringement because it would allow for infringement "based solely on some undefined 'need' to exchange data," and that the language did not limit the infringement in the presence of some anti-competitive activity.[101] The Japanese proposed rule changes also fail to define interoperability and could fall prone to an unintended broad exception and confusion as to what might constitute an infringing use versus a permitted use.[102]

B. Industry Solutions

1. Standards (Formal and Open)

The private sector has recognized for some time the advantages of creating standards and promoting interoperability through the formation of standard-setting organizations, patent pools, and licensing or cross-licensing agreements.[103] Formal standards can arise as the result of standard-setting organizations, government intervention, or more informally through the creation of patent pools in which firms recognize that they must work together to achieve business objectives and produce products that will succeed in the consumer marketplace.[104] When a formal standard is established, participants are often required to license the relevant patents on a reasonable and non-

[99] *See* FTC Report, *supra* note 13, at ch. 3, 51-53.

[100] *See* Sun Microsystems Position Paper, *supra* note 85.

[101] *See* USPTO Letter to European Commission, *supra* note 14, at 3. The USPTO also argued that too many uses would fall under this exception since "[v]irtually all computers exchange or are capable of exchanging data with other computers, and many require some 'conversion of conventions' for communication." *Id.*

[102] *See* METI Proposed Rules, *supra* note 94.

[103] For a discussion on standard-setting organizations, see Lemley, *supra* note 73. Examples of standard-setting bodies include W3C, IETF and OASIS.

[104] *See* Robert P. Merges, *Who Owns the Charles River Bridge? Intellectual Property and Competition in the Software Industry* 20 (Univ. Cal., Berkeley, Boalt Working Papers in Public Law, Paper No. 64, 1999), *available at* http://repositories.cdlib.org/cgi/viewcontent.cgi? article=1066&context=boaltwp.

discriminatory ("RAND") basis.[105] Access to these standards is usually readily available as a result of the threat of antitrust violations if the licensing is discriminatory.[106]

The private sector has also implemented other means to promote interoperability, such as open standards spearheaded by the open source movement.[107] Access to open source software is free, mandating only that additions and modifications to the software also be open source.[108] The private sector has at times recognized the overwhelming need for a free standard, such as the World Wide Web Consortium's ("W3C") decision to make Internet standards free rather than licensed under a RAND basis.[109] Additionally, some corporations have supported open standards by pledging free access to some patents within their patent portfolio.[110] IBM, who recently donated 500 patents to open standards, described its motives as aimed at promoting software interoperability. IBM stated that these patents "can contribute to open standards and broader interoperability between applications by providing open source developers with a solid base of innovation they can use and share" and that "[o]pen standards can accelerate the interoperability and expansion of the global infrastructure."[111] Perhaps a sign of a growing trend by the private sector toward promoting and achieving software interoperability, Microsoft also recently pushed the adoption of Open XML as an open international standard by which any Microsoft patent needed to implement any part of the XML specification would be freely available to anyone.[112]

[105] *See* Evans & Layne-Farrar*, supra* note 7*,* at 17.

[106] *See id.* at 40.

[107] *See id.* at 3-6, for a brief history of the open source movement.

[108] *See id.* at 12 (describing the GNU General Public License as "viral" since once the open source code is freely licensed, any other code that comes in contact with that code also then falls under the open license). Where a product contains part of a GPL licensed open source code, the GPL states that the creator "must cause any work. . .to be licensed as a whole at no charge to all third parties under the terms of this license." GNU General Public License, ver. 2, FREE SOFTWARE FOUNDATION, § 2(b) (June 1991), *available at* http://www.gnu.org/licenses/old-licenses/gpl-2.0.html.

[109] *See* Margaret Kane, *W3C Bows to Royalty-Free Pressure*, CNET, Nov. 14, 2002, http://news.com.com/2100-1001-965863.html. *See also* W3C PATENT POLICY, § 5.8 (Nov, 12, 2002), *available at* http://www.w3.org/Consortium/Patent-Policy-20040205/ [hereinafter *W3C Patent Policy*].

[110] *See* Press Release, IBM, IBM Pledges 500 U.S. Patents To Open Source In Support Of Innovation And Open Standards (Jan. 11, 2005), *available at* http://www-03.ibm.com/press/us/en/pressrelease/7473.wss.

[111] *Id.*

[112] *See* Microsoft Home Page, Interoperability, Choice and Open XML, http://www.microsoft.com/interop/letters/choice.mspx (last visited Mar. 24, 2008). ECMA International approved the adoption of Open XML on Dec 7, 2006 and has subsequently submitted it to the International Standards Organization (ISO) for its approval. *See* Press

2. Cooperative Disclosure

Besides standards, companies have also been known to disclose essential interfaces and protocols so that software developers and hardware manufacturers can write interoperable software.[113] As previously noted, software firms will often have an economic incentive to promote add-on applications because it usually makes their software more valuable to users.[114] Additionally, "[s]ome developers also provide access to their programs' underlying source code and combine these with opportunities for licensing so that users and others can take advantage of these disclosures for their own purposes."[115] Software firms also frequently enter into mutual cross-license agreements specifically aimed at achieving interoperability. This is often done with commercial interests in mind, such as attempting to bolster their products in the face of competing software applications.[116]

3. Why Current Industry Solutions are Insufficient

The success that the software industry has enjoyed as a result of standards, patent pools, licensing agreements, and open source does not mean that patents do not continue to have real potential to stifle innovation. Although high costs of entry related to inadvertent infringement might not be as pronounced in the interoperability context,[117] nevertheless, legal uncertainty and some costs of entry still exist and can act as significant barriers to innovation.[118] These

Release, Ecma Int'l, Ecma International Approves Office Open XML Standard (Dec. 7, 2006), *available at* http://www.ecma-international.org/news/PressReleases/PR_TC45_Dec2006.htm.

[113] *See* Smith & Mann, *supra* note 21, at 256.

[114] *See* Hart, Holmes, & Reid, *supra* note 8.

[115] Smith & Mann, *supra* note 21, at 256.

[116] *See* Press Release, Sun Microsystems, Microsoft and Sun Microsystems Enter into Broad Corporate Agreement; Settle Outstanding Litigation (Apr. 2, 2004), http://www.sun.com/smi/Press/sunflash/2004-04/sunflash.20040402.3.xml. The motivation behind the settlement and to working on interoperability between Windows and Sun's Solaris was aimed at defeating the Linux operating system's growing market share. *See* Brad Stone, *The Linux Killer*, WIRED, July 2004, http://www.wired.com/wired/archive/12.07/linux.html.

[117] *See* Leibovitz, *supra* note 12, at 2285. Traditional software firms would usually have to expend considerable resources on patent searches, investigations into the validity of patent uncovered, legal advice, and potentially defending lawsuits against patent holders. Even with a diligent patent search, the publication of a relevant patent may occur after an initial search, after a company has already spent a considerable amount of manpower and fiscal resources. *Id.* However, by the nature of the fact that a software firm is trying to interoperate with other software, these costs are likely reduced since that firm should be able to readily perform a focused search to know whether that technology with which it is attempting to interoperate is patented or not.

[118] *See* FTC Report, *supra* note 13, at ch. 3, 51-53 (finding that legal uncertainty is a

barriers to entry and legal uncertainty increase as computer programs grow more complex and potentially cause innovators to seek a growing number of licenses in order to achieve interoperability.[119]

Creating standards and licensing patents under a RAND basis has several potential drawbacks toward achieving software interoperability as well. For example, disputes over which technology to adopt as a standard provided the reasons, in part, for the opposition to XML as an open standard prior to its adoption in 2008.[120] Where the potential exists for dual standards to evolve, this can cause confusion and significantly derail the purpose for creating standards in the first place.[121] Patents on popular standards can also potentially provide broad monopoly power and create strong market positions for the holder of these patents, often a dominant player in the software industry. Since many firms participate in contributing patents to the standard, there is still some small risk of abuse in that "[f]ormal standards can still fall prey to the 'embrace, extend, and extinguish' strategy when a dominant firm is involved."[122] Other commentators note however that the growing complexity of software can decrease this potential for *de facto* standards and the ability of any one company from obtaining a complete monopoly since a controlling firm "will also have to deal with cross licenses, patent pools, and standard setting uses in its technologies that overlap with patent portfolios of other firms."[123] Additional drawbacks to standards are that standard setters often settle on vague guidelines regarding reasonableness,[124] and "after a standard has been

result of both the uncertainty as to the metes and bounds of a software patent claim and because of the questionable validity of a software patent as a result of low confidence in the quality of software patent examinations). The cost of entry can still be significant in the form of royalties, especially for situations involving interoperability where an entering firm has no choice presently but to license the technology from a willing patent holder or otherwise infringe that patent.

[119] *See id.* at ch.3, 44-45.

[120] *See* Kevin J. O'Brien, *Reversing Loss, Microsoft Wins Open-Format Designation*, N.Y. TIMES, Apr. 2, 2008, at C5 (commenting on the adoption of XML and touching upon the debate between adopting XML as an open standard where another previously approved rival standard already existed – the Open Document Format (OCF)).

[121] *See id.*

[122] Morgan, *supra* note 70, at 4. This strategy involves *embracing* compatible products or adoption of a standard, *extending* features of the tying software product or standard so as to create interoperability problems, and then *extinguishing* marginalized competitors once a *de facto* has been created. *See Microsoft: Deadly Embrace*, THE ECONOMIST, Apr. 1, 2000, at 58.

[123] Yang, *supra* note 72, at 187.

[124] See Evans & Layne-Farrar*, supra* note 7*,* at 17 (arguing that reasonable royalty is often left vague so as to avoid anti-trust issues and could be alleviated if antitrust regulators either encouraged more definite royalty rates or a process that standard setting bodies could follow to avoid anti-trust investigations).

adopted, what could be considered reasonable royalties of an essential component is quite different from what could be considered reasonable ex ante, when the component is still competing with other technologies."[125]

Interoperability also poses a unique problem in that software firms do not necessarily want to incorporate a patented technology or patented standard into their product in order to take advantage of that technology; rather, a firm often wants to create its own software technology while ensuring that consumers won't fall prey to network effects if the software was otherwise unable to communicate and interoperate with competing software or standards. As a result, tolls in the form of licensing fees merely to be able to interoperate can have the potential to discourage innovators from entering the market and can be leveraged by the dominant entrants to gain access to new patented technology merely based on a need to interoperate.[126] Even a reasonable royalty just to gain access to a standard for interoperability purposes can create a barrier to use and thus effectively reduce the number of available standard-supporting products.[127]

Open standards that employ royalty free licenses attempt to resolve the shortcomings of RAND, but open standards are not necessarily the answer to ensuring interoperability in all cases either. A royalty free license can "[push] away patent holders, who often are influential industry leaders. . .[and] also reduces the credibility of important standards because they are less likely to be adopted by those leaders,"[128] leading to the likelihood that a *de facto* standard – vulnerable to patent control – would then dominate.[129] Additionally, poor documentation of open source code has recently found some open source software battling patent infringement for inadvertently incorporating patented code into the open source.[130] Finally, although firms may have economic incentives to promote interoperability, some software firms have actively frustrated attempts to create interoperability with their products.[131]

[125] *See id.*

[126] Red Hat Inc., an open source advocate and service provider for Linux Software has made clear their concern that "[a] relatively small number of very large companies have amassed large number of software patents . . . [S]uch massive software patent portfolios are ripe for misuse." Red Hat Home Page, Statement of Position and Our Promise on Software Patents, http://www.redhat.com/legal/patent_policy.html (last visited Mar. 24, 2008).

[127] *See* Maxwell, *supra* note 71, at 127; *but see* Valimaki, *supra* note 83, at 7 (arguing that a reasonably priced interoperability licensing fee does not create barriers, but the licensing criteria can still "essentially close [off] interoperability information for those who cannot meet the licensing criteria," such as open source developers and academics that are often restricted to using only royalty free patents).

[128] Morgan, *supra* note 70, at 5.

[129] *Id.*

[130] *See* Letter from SCO Group, Open Letter to the Open Source Community (Sep. 9, 2003), *available at* http://www.sco.com/company/openletter/.

[131] *See* Deana Sobel, *A Bite out of Apple? iTunes, Interoperability, and France's Dadvsi*

As a result, standards and other industry solutions can still put up a large barrier to achieving interoperability. When software firms seek to incorporate a standard, a reasonable royalty may not be a high enough barrier to discourage innovation, but when the software merely seeks to interoperate with that standard, even a small amount of licensing fee may be enough to discourage firms from innovating and competing, and thus curtailing the efficiency and ease of use of products in the software industry.

C. Recent U.S. Case Law and the Potential Implications for Software Interoperability

1. The *eBay v. MercExchange* Decision

Although the recent Supreme Court case *eBay v. MercExchange* revolved around a business method patent and issuing permanent injunctions in patent infringement disputes, it may have a positive effect on software interoperability.[132] Prior to the *eBay* decision, a patent infringement case where the plaintiff prevailed almost always resulted in a permanent injunction.[133] The ability to prevent anyone from practicing an invention by threatening an injunction has a potential chilling effect on both innovators that need to infringe a software patent to achieve interoperability, and on licensing negotiations by giving the patent holders significant leverage. In *eBay,* however, the Court acknowledged with disapproval the growing industry of using patents to extort licensing fees and noted that the threat of an injunction "can be employed as a bargaining tool to charge exorbitant fees to companies that seek to buy licenses to practice the patent."[134]

The Court made clear that there should not be a general rule in patent cases that a finding of infringement and validity guarantees a permanent injunction against the infringer.[135] Rather, the Court upheld the traditional four-factor framework that requires a plaintiff to show:

> (1) that it has suffered an irreparable injury; (2) that remedies available at law, such as monetary damages, are inadequate to compensate for that injury; (3) that, considering the balance of hardships between the plaintiff and defendant, a remedy in equity is warranted; and (4) that the public

Law, 22 BERKELEY TECH. L.J. 267, 279-81 (2007) (commenting on Apple's iTunes protection software FairPlay that is specifically aimed at frustrating interoperability between other software and programs with iTunes).

[132] 547 U.S. 388 (2006). The case involved the accusation that eBay's "buy it now" application infringed MercExchange's one-click patent. The district court found infringement but granted damages and a compulsory license rather than a permanent injunction.

[133] *See id.* at 395 (2006) (Roberts, J., concurring).

[134] *Id.* at 396 (Kennedy, J., concurring).

[135] *See id.* at 392-93.

interest would not be disserved by a permanent injunction.[136]

In so holding, the Court effectively gave those seeking access to patented technology significantly more leverage in licensing negotiations. Addressing possible exceptions to the general rule, Justice Kennedy's concurrence also recognized that "[w]hen the patented invention is but a small component of the product the companies seek to produce and the threat of an injunction is employed simply for undue leverage in negotiations, legal damages may well be sufficient to compensate for the infringement and an injunction may not serve the public interest."[137] Software interoperability seems to fit nicely into this category since interoperability functions of a software invention should only comprise a very small amount of the total product.

2. Potential Effects of *eBay* on Software Interoperability

Although *eBay v. MercExchange* has yet to be applied where someone has infringed a patent in order to achieve interoperability, a court should be less likely to issue an injunction under the four factor test. The public interest prong of the four factor test weighs heavily in favor of software interoperability since courts have already recognized in the copyright context the strong public interest in interoperable computer software and the promotion of standards.[138] Further, the original patent holder may actually benefit from application software that interoperates and can still license the patented technology to other innovators. In a recent U.S. District Court decision *Paice LLC. v. Toyota Motor Corp.*, the court noted the small contribution that a patented hybrid technology made to the overall value of a motor vehicle and that the infringer did not directly compete with the patent holder. In that case, the court denied a permanent injunction,[139] but imposed a reasonable royalty on future infringement.[140]

However, because interoperability requires willful infringement, since the infringer knowingly uses patented software when reverse engineering, the possibility of treble damages still strongly deters innovators.[141] Alternatively,

[136] *Id.* at 391.

[137] *Id.* at 396-97 (Kennedy, J., concurring).

[138] *See* Sony Computer Ent't, Inc. v. Connectix Corp., 203 F.3d. 596, 602-04 (9th Cir. 2000).

[139] *See* Paice LLC v. Toyota Motor Corp., No. Civ. A. 04-211, 2006 WL 2385139, at *5 (E.D. Tex. Aug. 16, 2006) (holding that "[t]he infringed claims relate to the hybrid transmissions of the accused vehicles, but form only a small aspect of the overall vehicles").

[140] The court in *Paice LLC v. Toyota Motor Corp.* found that "[t]he jury, based on the entire record, determined an appropriate reasonable royalty rate that can be easily calculated on future sales of the accused devices thereby removing uncertainty from future damages calculations." *Id.*

[141] *See* 35 U.S.C. § 284 (2000) (allows treble damage awards for willful infringement). A recent U.S. District Court decision *Z4 Tech., Inc. v. Microsoft Corp.*, denied an injunction

if the reasonable royalty imposed by courts reflects only the small contribution that achieving interoperability represents to the entire software code, a resulting small royalty rate may make litigation economically impractical. If courts applied reasonable royalty rates in this way, the decreased threat of litigation would thus serve to significantly promote interoperability. Ultimately, this decision's impact on software interoperability still remains to be seen.

Thus, the answers to software interoperability within the patent system context are incompletely addressed by the international community, industry and U.S. case law. However, the insights gained by these proposals and partial solutions provide a valuable backdrop to finding a solution that effectively harmonizes promoting innovation with ensuring interoperability. With U.S. case law opening the door to flexible patent enforcement, some form of legislation seems necessary to complete the solution. Because the international community has shown their willingness to follow the U.S. in applying patent protection to software, the U.S. is uniquely situated to help resolve the international debate on software interoperability rather than merely relying on the precarious balance struck by the private sector.

V. SOLUTIONS FOR THE U.S. TO ENSURE CONTINUED INTEROPERABILITY AMONG FUTURE SOFTWARE INVENTIONS

A. *Complement Private Sector Solutions with Legislative Action*

1. Adopt a Well-Defined and Narrow Interoperability Exception to Patent Infringement

Academics have previously proposed exceptions to patent infringement for software interoperability within the U.S. Patent Law.[142] However, as the failure of the European proposal made clear, interoperability needs to be clearly defined in any proposed amendment to ensure sufficient continued protection for software patent holders against all other infringing uses. Considering that interoperability is not a novel concept to the software industry, a clear definition should be achievable if private industry input is sought when drafting a definition. Ensuring that an interoperability exception is not overly broad can also be achieved by narrowing the exception with such language as "indispensable" as proposed by Sun Microsystems,[143] and by

and granted a future reasonable royalty rate where infringement of a software patent was willful. *See* Z4 Tech., Inc. v. Microsoft Corp., 434 F. Supp. 2d 437 (E.D. Tex. 2006). However, this decision would seem to conflict with damages calculations for willful infringement.

[142] *See e.g.,* Smets & Pilch, *supra* note 12, at 29.

[143] *See* Sun Home Page, Position Papers: Software Patents, http://www.sun.com/aboutsun/policy/software_patents.html (last visited Mar. 24, 2008).

language found in METI's proposed rule changes that proposed looking at the disadvantage to society by enforcing such patent balanced against the benefits to the patent holder.[144] Alternatively, the best solution might be for Congress to leave this interpretation for the judicial system to create a definition over time through a case-by-case determination; however, such amendment in this form would seem unlikely to pass the scrutiny of the software industry and their lobbyists.

With some codified interoperability exception, even if significantly limited, the weaknesses seen in the industrial solutions will be substantially mitigated. *De facto* standards will have less of a stranglehold on products seeking to interoperate with that standard, and creation of *de facto* standards through "extend, embrace, extinguish" strategies will bear less risk to interoperability uses.[145] Additionally, the implications of choosing either a royalty free or a reasonable royalty for a standard would no longer affect users only seeking to interoperate with these standards by creating their own technology.

2. Amend the Patent Misuse Doctrine

Patent misuse is applied in rare situations where a patent holder has "impermissibly broadened the 'physical or temporal scope' of the patent grant with anticompetitive effect."[146] Often this situation arises where the patent holder is able to broaden the economic value from the patent by tying unpatented products to the patented product. Although this doctrine could potentially be interpreted to address interoperability concerns, refusal to license a patent is clearly specified as insufficient grounds for a finding of misuse.[147] Recent court decisions have significantly redefined the scope where patent misuse can be applied in the situation of extension of monopoly-type misuse by making clear that anticompetitive effect is a necessary element that must be proved by the infringer.[148] With this, proof of substantial market power becomes a necessary ingredient.[149] Thus, in the situation as it relates to interoperability, a patent holder would have to essentially be found guilty of some antitrust violation grounded in an anticompetitive context under a rule of

[144] *See* CompTIA Response, *supra* note 76, at 3.

[145] *See Microsoft: Deadly Embrace*, THE ECONOMIST, Apr. 1, 2000, at 58.

[146] Windsurfing Int'l, Inc. v. AMF, Inc. 782 F.2d 995, 1001 (Fed. Cir. 1986).

[147] *See* 35 U.S.C. § 271(d)(4) (2000).

[148] *See generally* Robert J. Hoerner, *The Federal Circuit and Antitrust: The Decline (And Fall?) of the Patent Misuse Doctrine in the Federal Circuit,* 69 ANTITRUST L.J. 669 (2002) (discussing the evolution of the patent misuse doctrine and the effect of monopoly type misuse that did not previously require anticompetitive effect).

[149] *Id.* at 684. The Supreme Court recently rejected the presumption of market power in product tying situations where a patent covers the tying product. *See* Illinois Tool Works v. Independent Ink, 547 U.S. 28, 44-45 (2006).

reason analysis.[150] Under the current U.S. case law and statutory code, it would seem that the patent misuse doctrine in its current form would rarely be applicable and unlikely to succeed except in the rarest of cases.[151]

Amendments to the patent misuse doctrine could be made by applying similar language of the Japanese proposed rule change.[152] Such language could find patent misuse where a license is refused when access is sought in order to achieve interoperability.[153] A finding of some specific intent or bad faith element would be required rather than the current requirement of a showing of market power. This change to the patent misuse doctrine and elimination of a market power requirement is justified because of the unique nature of the software industry that allows firms who hold patents on products that have effectively locked in consumers to essentially "broaden the 'physical or temporal scope' of the patent grant."[154]

This solution would be somewhat more limited in achieving interoperability as it would require a license fee in order to gain access to the patented technology. Although this may deter some would be innovators by increasing costs of entry, the decreased legal uncertainty may effectively balance these licensing costs. This solution would complement current industry solutions by ensuring access to technology where the patent holder is unwilling to license, as might be the case with a dominant firm with a *de facto* standard, or where otherwise the technology has yet to be made available through formal standards, patent pools, or open standards. Where patent pools and standards can often have significant lead times before they are made available, a patent misuse limitation would also improve accessibility to patent technology for interoperability purposes. Since the software industry is characterized by rapid change and first mover effects, this early access can have substantially positive effects on innovation and the public's interest in availability of interoperable software products.

B. Use Current and Future U.S. Case Law to Provide an Exception for Interoperability

1. Reinterpret Antitrust Case Law: The Essential Facility Doctrine

With the 2006 Supreme Court case *Illinois Tool Works v. Independent Ink* removing the presumption of market power in patent tying cases, it seems unlikely that the Supreme Court would read in a new exception for antitrust

[150] With no presumption of per se illegality or market power, the rule of reason analysis would create a significant burden of proof on the party alleging patent misuse.

[151] *See* Hoemer, *supra* note 148.

[152] *See* CompTIA Response, *supra* note 76, at 3.

[153] What would satisfy interoperability would also need to be clearly defined and could take a similar form as defined above.

[154] Windsurfing Int'l, Inc. v. AMF, Inc. 782 F.2d 995, 1001 (Fed. Cir. 1986).

tying cases as it relates to interoperable products without some Congressional action on this subject.[155] Additionally, antitrust laws have long recognized a company's fundamental right to refuse to deal with others.[156] However, one avenue of antitrust law that has yet to be explored by the courts is whether the refusal to license a patented technology for interoperability reasons could fall under the essential facility doctrine.[157]

The basic premise of the essential facility doctrine is that if a firm gains access to some facility unable to be replicated by a competitor, and uses it to ruin a competitor or put him at a significant disadvantage by not providing access to such facility,[158] then this is a violation of Section 2 of the Sherman Act if no legitimate business interest exists.[159] Interpreting this doctrine to allow for a software interoperability exception has several obstacles. First, the essential facility doctrine has been traditionally applied to tangible property.[160] Although intellectual property has never been held an essential facility, courts have, at least in several instances, conducted the analysis without distinguishing the fact that the claimed essential facility was intellectual property.[161] Second and most importantly, the Supreme Court has never

[155] *See* 547 U.S. 28, 44-45 (2006). Prior to this holding, a producer was presumed to have sufficient economic power in antitrust tying product cases when the tying product was covered by patent. *See* United States v. Loew's, Inc., 371 U.S. 38, 45 (1962).

[156] *See* Aspen Skiing Co. v. Aspen Highlands Skiing Corp., 472 U.S. 585 (1985). Although there are some limitations to this right, most notably the situation where a firm initially deals with a competitor and then at some point later refuses to deal without sufficient efficiency evidence, these limitations do not apply to situations where a firm is seeking to interoperate with no prior dealings with the patent holder. *See id.*

[157] For a detailed history of the evolution of the essential facility doctrine in the U.S., see David McGowan, *Regulating Competition in the Information Age: Computer Software as Essential Facility under the Sherman Act*, 18 HASTINGS COMM. & ENT. L.J., 771, 781-804 (1996).

[158] *See* Philip Areeda, *Essential Facilities: An Epithet in Need of Limiting Principles*, 58 ANTITRUST L.J. 841, 847-852 (1989). This principle has been defined by U.S. case law to compose 4 requirements: "(1) control of the essential facility by a monopolist; (2) a competitor's inability practically or reasonably to duplicate the essential facility; (3) the denial of the use of the facility to a competitor; and (4) the feasibility of providing the facility." MCI Communications Corp. v. AT&T Co*.,* 708 F.2d 1081, 1132-33 (7th Cir. 1983).

[159] 15 U.S.C.A. § 2 (2004).

[160] *See e.g.*, U.S. v. Terminal R. R. Ass'n of St. Louis, 224 U.S. 383 (1912) (controlling access to railroad terminals and the only bridges over a river); Otter Tail Power Co. v. U. S., 410 U.S. 366 (1973) (controlling access to power lines); Aspen Skiing Co. v. Aspen Highlands Skiing Corp., 472 U.S. 585 (1985) (controlling access to ski resorts).

[161] *See e.g.,* Intergraph Corp. v. Intel Corp., 195 F.3d 1346, 1356-58 (Fed. Cir. App. 1999) (court considered patented and propriety information under the essential facility doctrine without rejecting it merely because it was intangible property); Bellsouth Advertising & Pub. Corp. v. Donnelley Information, 719 F. Supp. 1551, 1566 (S.D. Fla.

formally recognized the essential facility doctrine.[162] However, in refusing to analyze *Verizon v. Trinko* under the essential facilities doctrine, the Court stated "where access exists, the doctrine serves no purpose."[163] Thus the Court left open the possibility of considering and recognizing the essential facility doctrine in a later case where an essential facility is not otherwise accessible.

Applying the essential facility doctrine thus remains a potential judicially enacted avenue to achieve software interoperability. Merely denying access to some patented software technology would not be enough to satisfy this doctrine.[164] The patented software must rise to the level of an essential facility, but this does not require that the facility be indispensable.[165] "To be 'essential' a facility need not be indispensable; it is sufficient if duplication of the facility would be economically infeasible and if denial of its use inflicts a severe handicap on potential market entrants."[166] However, antitrust violations via the essential facility doctrine have also usually been found where there is some form of natural monopoly – a situation where the relevant market can only support one firm since it would be otherwise unprofitable for competitors to replicate existing systems.[167] Although most interoperability needs would likely fail this natural monopoly requirement, in some situations such as where a *de facto* standard has been created through active acquisition of patented technology or where a company has implemented an "extend, expand, extinguish" strategy, it is very possible that a court could reasonably extend the essential facility doctrine to these cases and find that this is a form of natural monopoly.[168] Here, if a license is refused, the court could find a violation under Section 2 of the Sherman Act.[169] Expanding the current antitrust case law to these situations would provide a licensing solution to arguably the most

1988) (copyright infringement case where court held that "[a]lthough the doctrine of essential facilities has been applied predominantly to tangible assets, there is no reason why it could not apply, as in this case, to information.").

[162] *See* Verizon Comms. Inc. v. Law Offices of Curtis V. Trinko, LLP, 540 U.S. 398, 399 (2004) (recognizing the existence of the essential facilities doctrine but finding no need to either validate or repudiate it in the current case).

[163] *Id.* (finding that a government agency was already in place to ensure access to Verizon's network).

[164] *See* Berkey Photo v. Eastman Kodak, 603 F.2d 263, 281-82 (2d. Cir. 1979) (holding that Kodak was not required to disclose new film technology to a competitor in order for that competitor to make cameras that could take advantage of the benefits from this new type of film).

[165] *See* Hecht v. Pro-Football, Inc., 570 F.2d 982, 992 (D.C. Cir. 1977).

[166] *Id.*

[167] *See* McGowan, *supra* note 157, at 804-05 (noting that where there is no natural monopoly, then "the essential facility has no role to play because an entrant may replicate the facility in question.").

[168] *Id.*

[169] 15 U.S.C.A. § 2 (2004).

important of software interoperability needs.

2. Creating an Interoperability Exception to Willful Infringement

The potential implication of the *eBay* decision for software interoperability would be substantially expanded if firms seeking interoperability did not need to worry about treble damages from willful infringement. Congress is currently considering amendments to the Patent Act, which would include changes to what would constitute a finding of willful infringement.[170] These amendments would restrict a finding of willful infringement to three situations: infringement after specific notice from patent owner, copying with knowledge of patent and infringement after prior judgment.[171] Although these proposed amendments show Congress's willingness to limit the application of treble damages, infringing to achieve interoperability would still fall under the remaining current restrictions that the amendment proposes.[172] This is because a firm seeking interoperability will most likely know about the software patent since the patented technology will be necessary in order to achieve this compatibility. Although it seems unlikely that an additional interoperability exception would be incorporated into the current proposed amendments, if such reforms do pass through Congress under the second iteration of the Patent Reform Act, it does make future limitations to willful infringement more likely and feasible.

Such interoperability amendment to the willful infringement language would specifically exclude infringements aimed at achieving software interoperability. This exception would still protect the patent holder to recover normal damages,[173] while recognizing the strong public interest in promoting the development and innovation of interoperable software products. Additionally, the knowledge restriction would still remain intact as a means to treble damages, just with a small exception for interoperability purposes carved out. Since interoperability should be a relatively easy evidentiary burden to satisfy or refute, abuse to this exception to the knowledge requirement should be minimal. With this proposed change to a finding of willful infringement, achieving software interoperability would have yet another feasible and potentially broad avenue for relief within the judicial system.

[170] *See* Patent Reform Act of 2007, H.R. 1908, § 5, 110th Cong. (2007). Congress previously considered these amendments in much the same form under the Patent Reform Act of 2005, however that bill was never voted upon. *See* Patent Reform Act of 2005, H.R. 2795, § 6, 109th Cong. (2005).

[171] *See* Patent Reform Act of 2007, H.R. 1908, 110th Cong. § 5(c)(2) (2007).

[172] *See id.*

[173] *See* 35 U.S.C. § 284 (2000) ("[u]pon finding for claimant the court shall award the claimant damages adequate to compensate for the infringement").

VI. CONCLUSION

Since the early 1980's, the software patent domain has grown into a massive international playground complete with its share of "bullies." Although innovation relies on strong intellectual property rights, software interoperability is a justified need to warrant some limitations to these rights. Although the software industry has shown some initiative on its own to share,[174] often this is motivated by financial gain, and has potential shortcomings.[175] Both Europe and Japan have recognized this precarious balance that exists in the private sector and attempted to codify exceptions to software patent rights by recognizing software interoperability within their patent systems.[176] There is a strong public interest in promoting standards and software interoperability, and the U.S. is in a unique position to provide guidance to the international community.[177] The U.S. could take two possible legislative actions, such as allowing a specific exception for software interoperability similar to the CII Directive or modifying the patent misuse doctrine to recognize software interoperability. Additionally, expanding either the essential facility doctrine in antitrust law or the potential impact of the *eBay* decision by altering the requirements to what constitutes a finding of willful infringement through legislative amendments are two additional judicial avenues that could also promote software interoperability. On the software patent playground, a little paternalism will go a long way in creating a harmonious balance between the "bullies", older "kids", and the new "kids" that will create a healthier playground for everyone.

[174] *See* Merges, *supra* note 104.

[175] *See* Morgan, *supra* note 70, at 4.

[176] *See* CII Directive, *supra* note 15; Press Release, METI, *supra* note 17.

[177] *See generally* Park, *supra* note 5.