Draft as of August 21, 2008
Please do not cite, copy, or distribute without the author's permission.

Are Patents on Interfaces Impeding Interoperability?

by
Pamela Samuelson*

Abstract:

Many patents have issued for communications protocols and other interface designs for information and communications technologies (ICT). Commentators and policymakers have frequently expressed concern about the exclusionary potency of interface patents because they can confer on patentees power to block interoperability in socially harmful ways. This Article considers numerous policy options that have been proposed to respond to the dangers posed by interface patents, including exclusions of interfaces from patent protection, immunization of use of patented interfaces if necessary to achieve interoperability, withholding injunctive relief for infringement of interface patents, and treating refusals to license interface patents as abuses of intellectual property rights or violations of competition or antitrust laws. The Article observes that a great deal of interoperability is occurring notwithstanding the existence of interface patents, in part because private consortia have adopted policies in support of interoperability and in part because owners of such patents often have incentives to license them to facilitate interoperability. There is as yet insufficient evidence that interface patents are such serious impediments to interoperability as to justify adoption of strong measures, such as their exclusion from patent protection.

INTRODUCTION

Interoperability among information and communications technologies (ICT) is widely praised for promoting socially desirable goals, including fostering competition and innovation, enhancing consumer satisfaction, and promoting economic growth.[1] ICT interoperability means "the ability to transfer and render useful data and other information across systems, applications, or components."[2] To achieve interoperability, firms must have access to and be able to use the precise information that defines the

* Richard M. Sherman Distinguished Professor of Law and Information, University of California, Berkeley. I wish to thank Tom Kearney for his excellent research assistance for this article and Robert Barr, Michael Carrier, Rochelle Dreyfuss, Joe Farrell, Robert J. Glushko, Bernt Hugenholtz, Mark Lemley, Polk Wagner, Phil Weiser, Christopher Yoo, and various software industry executives for their useful comments on earlier drafts of the article.

[1] See, e.g., URS GASSER & JOHN PALFREY, BREAKING DOWN DIGITAL BARRIERS: WHEN AND HOW ICT INTEROPERABILITY DRIVES INNOVATION (Nov. 2007) at 1, available at http://cyber.law.harvard.edu (hereinafter "ICT Interoperability"). Gasser & Palfrey observe that it is very difficult to find anyone who speaks out against interoperability. Id.

[2] Id. at 4. See also INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. IEEE STANDARD COMPUTER DICTIONARY: A COMPILATION OF IEEE STANDARD COMPUTER GLOSSARIES (1990) (interoperability defined as "the ability of two or more systems to exchange information and use the information exchanged").

boundaries between ICT systems, that is, the interfaces between them. Insofar as patents are issuing on interface designs and components, they would seem to present potential impediments to interoperability. This Article considers whether such patents are, in fact, impeding interoperability, and if so, what should be done about it.

Part I discusses explains how intellectual property (IP) rules have evolved to enable firms to protect interfaces and reviews some complex and dynamic factors that firms consider in deciding whether to seek patents for ICT interfaces. It gives some examples of interface patents that have had an impact on interoperability.

Part II discusses the extensive array of policy options that commentators, policymakers, and courts have suggested as possible responses to the exclusionary potential of ICT interface patents. Subpart A considers proposals that would, in essence, make interfaces unprotectable by patent law.[3] Subpart B discusses some ways in which patent rules might be tailored to facilitate interoperability. Subpart C focuses on some private initiatives, including some undertaken by standard-setting organizations (SSOs), aimed at controlling the unbridled exclusionary power of interface patents by requiring commitments to license such patents, insofar as they are essential to achieving interoperability, on royalty-free (RF) or reasonable and non-discriminatory (RAND) terms. Subpart D explores some proposals to subject interface patents to liability rules, as by withholding injunctive relief against those who use patented interfaces to achieve interoperability. Subpart E considers competition law as a source of oversight of a dominant firm's refusal to license interface information and IP rights (if any) in such information. In particular, it will review the European Commission's order requiring Microsoft Corp. to prepare documentation of its interfaces and to make this documentation available on reasonable licensing terms to competitors in the work group server operating system (OS) market.[4]

Part III observes that there is a considerable amount of interoperability in today's ICT environment, notwithstanding the issuance of interface patents. At present, there seem to be sufficient incentives for firms to make interface information available or license interface patents so that stronger measures, such as excluding interfaces from patent protection or immunizing use of interface patents to achieve interoperability, do not seem to be warranted. Interface patents pose the gravest risks for competition and follow-on innovation when practice of such patents are essential to interoperability, when the patents are held by established firms with market power, and when there are incentives for firms to enforce interface patents in a manner that provides the opportunity for leveraging a dominant firm's power in one market into that of an adjacent market.

---

[3] Part II-A offers an interpretation of the European Directive on the legal protection of computer programs as excluding interfaces from IP protection. See Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs, O.J. L 122, 17/05/91, p. 42 (hereafter "Software Directive").

[4] See European Commission Decision 2007/53/EC relating to a proceeding pursuant to Article 83 [EC] and Article 54 of the EEA Agreement Against Microsoft Corp., Case COMP/C-3.37.792—Microsoft), OJ 2007 L 32, p. 23 (March 24, 2004) (hereafter "EC Microsoft Decision"). The Commission's order was affirmed in Microsoft Corp. v. Commission of the European Communities, Case No. T-201/04, Court of First Instance, Sept. 17, 2007 (hereafter "CFI Microsoft Decision").

The appropriate focus for regulation of interface patents should be on these circumstances, rather than on interface patents as such.

I. The Role of IP Law in Protecting Interfaces

This Part considers the role that intellectual property rights (IPRs), and patents, in particular, have been playing in the protection of interfaces of ICT systems and the effects of IP rights on the ability of others than the interface's developer to achieve interoperability. To set the stage for this discussion, it is useful first to offer some more refined definitions of interfaces and interoperability.

A. Some Definitions and Preliminary Observations

The International Organization on Standards defines interoperability as "[t]he capacity to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units."[5] A recent book has explained:

> Interoperability doesn't require that two systems be identical in design or implementation, only that they can exchange information and use the information that they exchange. Interoperability requires that the information being exchanged is conceptually equivalent: once this equivalence is established, transforming different implementations to a common exchange format is a necessary but often trivial thing to do.[6]

Interoperability is enabled when the maker of one ICT system develops interfaces to enable the exchange of information between the entity it is developing and the entities with which its entity will interact.[7]

One of the principal reasons that modern ICT systems are so powerful is because they can often call upon other systems' functionalities. It is not necessary, for example, for each software developer to write code to perform common functions, such as accessing files or directories, because developers of operating systems (OS) have incorporated these functionalities into their systems. A platform developer that wants others to build applications for that platform (which they often do in order to make their platform attractive to consumers) must make application programming interfaces (APIs) available to other firms. APIs disclose the "hooks" (aka standardized requests) that other

---

[5] ISO/IEC 2382-01, IT Vocabulary, Fundamental Terms.

[6] ROBERT J. GLUSHKO & TIM MCGRATH, DOCUMENT ENGINEERING: ANALYZING AND DESIGNING DOCUMENTS FOR BUSINESS INFORMATICS AND WEB SERVICES 172 (2005).

[7] ICT interfaces are informational equivalents of the standard plug and socket designs that designers of appliances must use in order for their appliances to successfully interoperate with the electrical grid for which they are designed. See, e.g., Pamela Samuelson, *The Strange Odyssey of Interfaces in Intellectual Property Law*, in CON/TEXTS OF INVENTION (Mario Biagioli, et al., eds. forthcoming 2008) (making this analogy). Although all countries have standardized on some electrical socket and plug designs, many countries have standardized on different socket and plug designs; this is why international business travelers have to bring multiple plug kits, as well as transformers, with them when they travel.

developers must use to call upon the platform's services to carry out specific tasks for the applications. In this respect, APIs are one-sided and outward-facing. That is, the developer of the API doesn't need to reveal to other developers the fine details about how it provides the relevant service; it only needs to supply the API which defines the manner in which to request and successfully invoke the platform's services.

While APIs typically define the specifications for information exchanges between a platform and applications built on it, protocols are also important components of interfaces because they facilitate communications (i.e, interoperability) across different computer or ICT systems. Protocols define rules for information exchanges by specifying, for example, how to start and end messages, how to format messages, what to do with corrupted or improperly formatted messages, and the like. ICT systems typically involve multiple layers of functional units that interact with one another through a set of defined protocols.[8] Many ICT protocols have become widely used standards (e.g., SMTP and HTTP).

An important distinction is that between interfaces and implementations.[9] Program interfaces are abstractions of the services they are designed to invoke. However, interfaces can only be effective in calling upon those services when implemented in source and object code. Programmers can design program internals to implement interfaces and encode those designs in source code in many different ways. The interface does, however, constrain program design to some degree because of the need to precisely conform a request for the program's services to the interface specification for those services.

Although it is useful to conceptualize interoperability at a high level of abstraction, it is important to realize that interoperability has somewhat different meanings in different ICT contexts.[10] In the context of computer programs, for example, it means that programs can function effectively with other software and/or hardware to carry out the tasks they were designed to perform.[11] In the context of digital identification systems, interoperability means the ability of users to sign on to one service and have their personal data transferred securely to those with whom the users are transacting.[12] In the context of technically protected digital music, interoperability means

---

[8] An important principle for modern communications systems is that "the entity responsible for a given protocol should respond only to events and messages from its counterpart in the same layer at the other end of the communication." Id. An email server, for instance, can and should signal receipt of a message from another email server, but not from other applications in different layers of the stack. Id.

[9] See, e.g., Alfred Z. Spector, *Software, Interface, and Implementation*, 30 Jurimetrics J. 79 (1989).

[10] See, e.g., ICT Interoperability, supra note 1, at 4-5.

[11] "The function of a computer program is to communicate and work together with other components of a computer system and with users and for this purpose, a logical and where appropriate physical interconnection and interaction is required to permit all elements of hardware and software to work with other software and hardware and with users in all the ways in which they are intended to function." This functional interconnection and interaction is what the Directive characterizes as interoperability. Software Directive, supra note 3, recitals 10-11.

[12] ICT Interoperability, supra note 1, at 5. See also JOHN PALFREY & URS GASSER, CASE STUDY: DIGITAL IDENTITY INTEROPERABILITY AND EINNOVATION, Nov. 2007, available at http://cyber.law.harvard.edu/interop;

that the music can be played on many devices and made available in a variety of online channels.[13] In the context of electronic commerce, interoperability means an exchange of messages (e.g. an order and an acceptance of an order) that will result in a successful business transaction.[14] Although this article will mainly focus on computer program interfaces, many of the same legal and policy issues, as well as technical, economic, and business issues affecting interoperability cut across ICT systems.

Many stakeholders have interests in interoperability. Developers of platforms have a very big stake in interoperability because they benefit from the development of applications that work on their platforms. This is largely because of the positive feedback loop created by network effects, as customers are drawn to the platform as more applications are available for the platform, and more applications developers are drawn to the platform as the platform attracts more customers.[15] Developers of these complementary products benefit by interoperability when they create products that work on platforms with large customer bases. Consumers benefit from interoperability because they can use the same information resources on multiple platforms in a "plug and play" fashion.[16] Many intermediaries, such as vendors of ICT products, benefit when interoperability exists among products in the marketplace, as it is easier to sell compatible components. Other platform developers may seek to develop platforms compatible with popular platforms, arguing that this creates a level playing field on which competition can occur based on price, quality, and differences in feature sets. The market may become larger for all players when there is one interface and many implementations, rather than multiple platforms, each of which is non-interoperable with the other. The interests of successful platform developers and emerging competitors may, however, not be aligned.

Interoperability is often conceived as a binary concept: one ICT entity either interoperates with another ICT entity or it doesn't. From the users' standpoint, there is certainly something to this. But interoperability is more appropriately regarded as a continuum or spectrum,[17] along which some entities (e.g., programs or content) are more interoperable than others. At one end of the spectrum are entirely closed systems that reveal no APIs; at the other end of the spectrum are systems that expose all of the details of its design, including APIs, such as open source software.

Microsoft is somewhere in the middle of this spectrum. It is closer to the open end of this spectrum insofar as it publishes many of its APIs and licenses others. These APIs are generally sufficient to allow independent software vendors (ISVs) to write programs that will operate on Windows-based platforms.[18] Microsoft does not, however, disclose all of the interface information that ISVs might want to know. Often the

---

[13] ICT Interoperability, supra note 1, at 5. See also URS GASSER &JOHN PALFREY, CASE STUDY: DRM-PROTECTED MUSIC INTEROPERABILITY AND EINNOVATION, Nov. 2007, available at http://cyber.law.harvard.edu/interop.
[14] Glushko & McGrath, supra note 6, 172-80.
[15] See, e.g., Mark A. Lemley & David McGowan, *Legal Implications of Network Effects*, 86 Cal. L. Rev. 479 (1998); Joseph Farrell, *Standardization and Intellectual Property*, 30 Jurimetrics J. 35 (1989).
[16] See, e.g., Jonathan Zittrain, *The Generative Internet*, 119 Harv. L. Rev. 1974 (2006).
[17] ICT Interoperability, supra note 1, at 4.
[18] Many of Microsoft's APIs are published at http://www.microsoft.com.

undisclosed information pertains to how one component of the platform calls upon the services of another component.[19] Yet, having greater access to information about these interfaces may enable ISVs, including open source developers, to achieve better performance or build a richer feature set for their programs.[20] Frustration over limits on Microsoft's disclosures of internal APIs has caused some to engage in reverse analysis to discern and document unlicensed interface information.[21]

Whether firms such as Microsoft should be required to disclose interface information to other firms to help them develop interoperable systems has been a contentious issue in recent years. Microsoft argued to the European Commission that it had no legal obligation to supply Sun or other makers of work group server (WGS) OS systems so that they could develop functionally equivalent programs to Microsoft's WGS-OS. Microsoft argued further that its IPRs in program interfaces justified its refusal to provide the information sought by its competitors.

Why do ICT firms, such as Microsoft, sometimes adopt a business strategy that depends on controlling interoperability? It is because this strategy has the potential to be more lucrative than a completely open, fully interoperable strategy would be. The successful developer of a product that controls interoperability may enjoy large profits that do not need to be shared with others. When Apple launched its iTunes service for selling digital music to customers of its iPod technology, for example, it hoped to establish its own network and network effects without direct competition from other music platform providers. Apple's considerable success with this strategy gave RealNetworks incentives to reverse engineer Apple's FairPlay technology so that it could make its RealPlayer compatible with iTunes music. Although Apple's initial response was to threaten to sue RealNetworks for violating certain IP rules, it soon adopted a more effective response by changing the iTunes interface to disable the RealPlayer's compatibility feature.[22] As the Microsoft and Apple examples illustrate, IP rights may be an important means by which firms with controlled or non-interoperable business strategies can stop unlicensed persons from interoperating with their platform.

B. Evolving IP Rules and Practices as to Interfaces

---

[19] Although Microsoft may justify non-disclosure on the grounds that it did not consider such information to be part of the APIs that ISVs need to know, critics of Microsoft have sometimes charged it with hiding interface information in order to gain strategic advantages (e.g., faster implementations of certain key functions) over firms whose products compete with Microsoft's. See, e.g., Groklaw, *Microsoft's Allegedly Undocumented APIs*—Comes v. Microsoft, Feb. 8, 2007, available at http://www.groklaw.net/articlebasic.php?story=2007020819534335.

[20] Whether undocumented details are "essential" to achieving meaningful interoperability is likely to be a matter of some debate. See, e.g., Nokia Corp. v. Interdigital Techn. Corp., [2007] EWHC 445 (Pat) (addressing dispute between the parties as to the "essentiality" of certain patents to compliance with standards adopted by the European Technical Standards Institute for mobile telephones).

[21] Numerous books have disclosed such APIs. See, e.g., SVEN SCHREIBER, UNDOCUMENTED WINDOWS 2000 SECRETS: A PROGRAMMERS' COOKBOOK (2001). While some programmers believe that more API information is always better, there are some costs associated with extensive APIs. The more extensive they are, the more difficult it may be to learn and make good use of them.

[22] Gasser & Palfrey, supra note 1, at 1.

In the early years of computing industry (i.e., before the mid-1970's), developers of hardware and software, including major firms such as IBM, often distributed source code and interface specifications without IP restrictions.[23] Firms had incentives to make source code and/or interface specifications available and allow unrestricted use of them so that customers could, for instance, customize the technologies to meet their needs and so that other firms could make complementary products that would work on the hardware or with the software installed on that hardware. Even before the term "network effects" was coined to describe the phenomenon, it was obvious that a firm could create demand for its platform by aiding others to develop information resources for it.[24]

Starting in the mid- to late 1970's, it became more common for firms to withhold source code or interface information from those who might want access to them. Commercial software developers began to think of source code and interface specifications as trade secrets. They began distributing programs in object code form and claiming copyright in that code. They often hoped that copyright would not only protect object code against exact copying, but would, in conjunction with anti-reverse engineering clauses in software licenses, protect the developer against reverse engineering (which inevitably requires making intermediate copies of the code), thereby indirectly preventing any trade secrets in their interfaces from being discerned and thwarting the efforts of unlicensed parties to make interoperable systems.[25]

In the mid-1980's to the mid-1990's, some firms, IBM prominently among them, also argued copyright protection should be available for original interfaces embodied in programs.[26] The issue first arose in a lawsuit that challenged the copyrightability of the

---

[23] See, e.g., JONATHAN BAND & MASANOBU KATOH, INTERFACES ON TRIAL 19 (1995). See also Anita Stork, *The Use of Arbitration in Copyright Disputes*: IBM v. Fujitsu, 3 High Tech. L. J. 241 (1987) (pointing out that IBM distributed source code without copyright restrictions through the 1970's).

[24] The open publication strategy of that era may also have been affected by uncertainties that then existed about whether computer programs, let alone interfaces, qualified for either copyright or patent protection. Although the Copyright Office began accepting registrations of computer programs in 1965, it did so under its "rule of doubt;" indeed, the registration certificates indicated the Office's doubt about the copyrightability of programs in machine-executable form. See, e.g., Copyright Office Circular 31-D (1965), reprinted in Duncan M. Davidson, *Protecting Computer Software: A Comprehensive Analysis*, 1983 ARIZ. ST. L.J. 611, 652 n.72. See generally Pamela Samuelson, *CONTU Revisited: The Case Against Copyright Protection for Computer Programs in Machine-Executable Form*, 1984 Duke L. J. 663. Doubts about the patentability of programs arose because programs are texts and because many information innovations embedded in programs, such as algorithms, are "mental processes" (that is, processes that can be carried out in the human mind or with the aid of a pen and paper). See, e.g., Gottschalk v. Benson, 409 U.S. 63 (1972) (denying patentability of algorithm for transforming binary coded decimals to pure binary form). See generally Pamela Samuelson, Benson *Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 Emory L. J. 1025 (1990) (discussing case law and doctrinal developments).

[25] The core argument for this approach is discussed in Allen R. Grogan, *Decompilation and Disassembly: Undoing Software Protection*, Computer Law., Feb. 1984, at 1.

[26] Trademark and false advertising law has sometimes been used to challenge those who have developed compatible ICT systems. See, e.g., Creative Labs, Inc. v. Cyrix Corp., 42 U.S.P.Q.2d 1872 (N.D. Cal. 1997)(granting preliminary injunction against sales of sound cards that were not "fully compatible" with the plaintiffs' system, as its ads claimed).

Apple II OS programs.[27] Makers of Apple clone computers claimed, among other things, that it was necessary to copy the Apple OS so that their work-alike computers could achieve interoperability with programs written for the Apple platform.[28] One court responded by saying:

> Franklin may wish to achieve total compatibility with independently developed application programs written for the Apple II, but that is a commercial and competitive objective which does not enter into the somewhat metaphysical issue of whether particular ideas and expressions have merged.[29]

The dicta from this decision cast a shadow over the prospects for future defenses to copyright claims based on the copying of interfaces.

This dicta was given a further boost in Whelan Associates, Inc. v. Jaslow Dental Lab., Inc.[30] *Whelan* characterized computer programs as "literary works" and reasoned that since copyright law had long protected non-literal elements (i.e., structure and organization) of literary works, such as novels and plays, it should protect the structure, sequence, and organization (SSO) of programs as well.[31] *Whelan* deemed all program SSO to be protectable by copyright law as long as there was more than one way to structure a program to achieve the program's functions.[32] If there was just one way to structure a program to perform particular functions, though, the "idea" of that function and its structural "expression" would be "merged" and treated as among the unprotectable program "ideas."[33] Without broad copyright protection for computer programs, and in particular, for aspects of program SSO that were costly and difficult to develop as well as commercially significant, the court worried that there would be too little protection to provide proper incentives to develop computer programs.

The *Whelan* case did not involve claims of copyright infringement as to interfaces necessary for interoperability, but the analysis in *Whelan* seemed to support such claims. Relying heavily on *Whelan*,[34] Computer Associates sued Altai for infringement for

---

[27] See Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240 (3d Cir. 1983); Apple Computer, Inc. v. Formula Int'l, Inc., 725 F.2d 521 (9th Cir. 1984). These lawsuits were somewhat surprising, given that Congress had amended copyright law to clarify that programs could be copyrighted. Pub. L. No. 96-517, 94 Stat. 3007, 3028 (codified at 17 U.S.C. §§ 101, 117 (1982)).

[28] *Franklin*, 714 F.2d at 1245-46. Franklin also argued that the Congress had only intended to protect application programs that interacted with people, not purely functional programs such as operating systems. Id. at 1246-52. The court did not find this or other Franklin defenses persuasive. Id. at 1251.

[29] Id. at 1253.

[30] 797 F.2d 1222 (3d Cir. 1986). For a detailed explanation of the flaws in the *Whelan* analysis of the scope of copyright in computer programs, see, e.g., Pamela Samuelson, *Why Copyright Excludes Systems and Processes From the Scope of Its Protection*, 85 Tex. L. Rev. 1921, 1962-73 (2007).

[31] *Whelan*, 797 F.2d at 1234.

[32] Id. at 1236.

[33] Id. at 1228, 1247.

[34] See, e.g., Reply Brief for Plaintiff-Appellant, in Computer Associates Int'l, Inc. v. Altai, Inc., 1991 WL 11010234 (relying heavily on *Whelan*).

copying of CA's interfaces,[35] pointing to substantial similarities between the compatibility components of Altai's Oscar program and its CA-Scheduler program, especially as to their parameter lists (i.e., lists of information that must be sent or received by subroutines to invoke specific scheduling tasks). CA argued the parameter lists had been carefully and precisely designed, making them costly to develop and commercially significant parts of programs. CA argued that incentives to invest in software development would be undermined if competitors such as Altai could appropriate program SSO without fear of liability. Parameter lists and other SSO elements of program interfaces are, moreover, complex and detailed, which under *Whelan* made them protectible expression. In view of the SSO similarities, CA argued the revised Oscar still infringed the copyright in CA-Scheduler.

Altai, however, persuaded the court to recognize that external factors may constrain the design choices of programmers.[36] Because CA-Scheduler and Oscar provided the same scheduling services and both were designed to interoperate with same IBM OS programs, similarities in their parameter lists were understandable and not evidence of infringement.[37] The court in *Altai* asserted that extending copyright protection to program interfaces would "have a fundamentally corrosive effect on certain fundamental tenets of copyright doctrine"[38] and suggesting patents as an alternative form of IP protection.[39]

The *Altai* decision may not initially have caused software developers and their lawyers to think seriously about patenting interfaces and other program SSO, in part because it took some years for *Altai* to defeat *Whelan* in the subsequent caselaw and emerge as the leading decision about software copyrights.[40] However, the patent option became more urgent after the Ninth Circuit Court of Appeals in California issued its ruling in Sega Enterprises, Ltd. v. Accolade, Inc.[41]

Sega was important in the IP-in-interfaces saga for at least four reasons. For one thing, it embraced *Altai*'s rhetorical approach to conceptualizing computer programs as utilitarian works eligible for only a thin scope of copyright protection.[42] Second, *Sega* followed *Altai* in ruling that program interfaces were elements of programs that copyright

---

[35] Computer Associates Int'l, Inc. v. Altai, Inc., 775 F. Supp. 544 (E.D.N.Y. 1991), aff'd, 982 F.2d 693 (2d Cir. 1992).

[36] The court relied on the Nimmer treatise which had taken the position that interfaces were aspects of programs for which no copyright protection should be available because of the constraints they placed on design choices of subsequent programmers. See MELVILLE D. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT, sec. 13.03, cited in *Altai*, 982 F.2d at 709-10.

[37] See Brief of Defendant-Appellee, Computer Associates Int'l, Inc. v. Altai, Inc., 1991 WL 11010232 (making this argument).

[38] *Altai*, 982 F.2d at 712. The court criticized *Whelan* for its unduly broad conception of the scope of copyright in computer programs, for its reliance on metaphysical distinctions rather than practical considerations, and for its outdated comprehension of computer science. Id. at 705-06.

[39] Id. at 712.

[40] *Altai* has been followed in at least 49 subsequent decisions.

[41] 977 F.2d 1510 (9th Cir. 1992).

[42] See, e.g., id. at 1526 ("Under the Copyright Act, if a work is largely functional, it receives only weak protection.")

law did not protect; indeed, *Sega* spoke of interface information as "functional requirements for achieving compatibility with other programs."[43] Third, the court ruled that copying program code in the course of reverse engineering it for a legitimate purpose such as extracting interface information to make a compatible program did not infringe copyright in that code.[44] The court recognized that

> If disassembly of copyrighted object code is per se an unfair use, the owner of the copyright gains a de facto monopoly over the functional aspects of his work—aspects that were expressly denied copyright protection by Congress. In order to enjoy a lawful monopoly over the idea or functional principle underlying a work, the creator of the work must satisfy the more stringent standards imposed by the patent laws.[45]

Fourth, it indicated that even copying some exact code from another program would not be infringement insofar as that code was essential to achieving interoperability.[46]

After *Sega*, developers could no longer hope to protect interfaces by copyright. Because *Sega* endorsed unlicensed copying of code to extract interface information,[47] it imperiled developer efforts to protect interfaces as trade secrets. *Sega* signaled that the only reliable means for protecting the functional requirements for achieving interoperability was by patenting them. Patents had a key advantage over copyrights in protecting interfaces because patent law has no "merger" doctrine. Hence, if there is only one way to achieve a particular function and a developer has patented that one way, it can enforce its patent to stop unlicensed uses.[48] Moreover, patent law also has no explicit reverse engineering privilege.[49]

Altai and *Sega* contributed to the eventual shift away from claims of copyright in program interfaces and toward reliance on patent protection. Patent protection also became more plausible and attractive as courts became more receptive to software patents.[50] The Supreme Court had initially cast doubt on the patentability of program

---

[43] Id. at 1525-26.
[44] Id. at 1527-28 (finding the reverse engineering copies to be fair use).
[45] Id. at 1525.
[46] Id. at 1516. See also id. at 1528-32 (treating some Sega code too functional for IP protection).
[47] Prior to *Sega*, some commentators had argued that reverse engineering of object code should be treated as both copyright infringement and trade secret misappropriation, copyright infringement because of the intermediate copying required to reverse engineer and trade secret misappropriation because the infringing copies made in reverse engineering constituted an improper means to get the trade secrets embodied in the object code. See, e.g., Allen Grogan, *Decompilation and Disassembly: Undoing Software Protection*, COMPUTER LAW., Feb. 1984, at 1.
[48] Patent law has, however, some policy levers that can be employed to limit the scope of patents. See, e.g., Dan L. Burk & Mark A. Lemley, *Policy Levers in Patent Law*, 89 Va. L. Rev. 1575 (2003).
[49] But see, e.g., Julie E. Cohen & Mark A. Lemley, *Patent Scope and Innovation in the Software Industry*, 89 Cal. L. Rev. 1 (2001) (exploring patent doctrines that could be interpreted to permit reverse engineering of software even if some components of the software were patented).
[50] The European experience with software patents and special concerns about interface patents are discussed infra notes xx and accompanying text. A concise history of European perspectives on software patents can be found in a study commissioned by the European Parliament about the patentability of computer programs in connection with its consideration of a Proposal for a Directive of the European

innovations in the 1970's,[51] but the Court was receptive to one such patent in its 5-4 decision in 1981 in Diamond v. Diehr.[52] Relying on some broad language in *Diehr*,[53] the Federal Circuit during the 1980's and 1990's developed a capacious conception of patentable subject matter under which virtually all computer program-related inventions are patentable.[54] This, coupled with increasing "thinness" of copyright protection after *Altai* and *Sega* achieved widespread acceptance in the mid-1990's, led to big surge in patenting of software innovations,[55] including more issuance of patents on interfaces.

Although program interfaces seem patentable subject matter in the U.S.,[56] relatively few interfaces are, in fact, patented. Choosing not to patent interfaces makes sense for developers whose business strategy relies upon publication of interfaces as a potential generator of network effects, as well as for interfaces developed through collaborations among industry representatives and interfaces for open source projects.

Even firms with a more proprietary approach toward their interfaces may still have good reasons not to patent them. For one thing, program interfaces can often be protected quite effectively as trade secrets. Because commercially distributed programs are typically shipped in machine-executable form, not in human-readable form, program interfaces are not readily discernible when running the program through its various tasks. Trade secrecy is a much cheaper and easier means of getting IP protection for an interface than seeking a patent; it also obviates the need for any disclosure of any innovation the interface embodies.

Trade secrecy can, of course, be jeopardized by reverse engineering conducted by those who want to get access to interface information, but firms can and often do counteract this risk by inserting anti-reverse engineering clauses into their license agreements and/or by devising strategies for making interfaces difficult to discern.[57] The more complex a program is, moreover, the more difficult it will be to reverse engineer to

---

Parliament and of the Council on the Patentability of Computer-Implemented Inventions, COM(02)92 final, available at http://www2.europarl.eu.int/oeil/file.jsp?id=219592 (hereinafter "Proposed Software Patent Directive"). See Reinier Bakels & P. Bernt Hugenholtz, *The Patentability of Computer Programs: Discussion of European-level Legislation in the Field of Patents for Software*, Legal Affairs Series, JUI 107 EN, 04-2002, available at http://www.ivir.nl/publications/other/softwarepatent.html.

[51] Gottshalk v. Benson, 409 U.S. 63 (1972); Parker v. Flook, 437 U.S. 584 (1978).

[52] 450 U.S. 175 (1981). Diehr claimed a new method of curing rubber that used a computer program to calculate when the temperature of the rubber inside molds had reached the proper curing point. The PTO rejected Diehr's claim because its only novelty lay in the program's calculations. The *Diehr* decision was initially perceived as a modest change in the patent landscape as to program-related inventions because the Court was so deeply divided, because the majority opinion did not repudiate the Court's earlier rulings on the unpatentability of certain program innovations, and because *Diehr* involved a traditional manufacturing process (i.e., curing rubber),.

[53] *Diehr*, 450 U.S. at 181 (patentable subject matter includes everything under the sun made by man).

[54] See, e.g., AT&T Corp. v. Excel Comm'ns, Inc., 172 F.3d 1352 (Fed. Cir. 1990).

[55] See Josh Lerner & Feng Zhu, *What is the Impact of Software Patent Shifts? Evidence from* Lotus v. Borland 10 (Nat'l Bur. Econ. Res. Working Paper No. 11168 2005) (empirical evidence of surge in patenting of software dating from the mid-1990's).

[56] See infra Part II-A for a discussion of possible rationales for excluding interfaces from patent protection.

[57] This is a common practice in the software industry. See, e.g., Pamela Samuelson & Suzanne Scotchmer, *The Law and Economics of Reverse Engineering*, 111 Yale L.J. 1575, 1626-30 (2002).

get access to interfaces. If unlicensed parties successfully reverse engineer a firm's interface, the firm whose products have been reverse engineered can, moreover, change the interface in subsequent versions of the program, thereby impeding interoperability by unlicensed firms. Still, some interfaces are patented, so it is worth considering some reasons why patenting interfaces sometimes makes sense.

C. Incentives for Patenting Interfaces

The main reason why firms seek patents for interfaces is because such patents can be very powerful in conferring an exclusive right to control the development not only of competing but also of complementary products insofar as the interface defines the boundaries between ICT systems. [58] It is, moreover, easy to detect infringement of interface patents because if unlicensed products successfully interoperate with the patentee's products, they almost certainly infringe.[59]

Interface patents are also valuable because it may be impossible to work around them. Even a very narrowly drawn interface patent may preclude interoperability as to key functions.[60] The exclusionary power of interface patents is, moreover, strong even if the technical design disclosed in the patent is only modestly or even trivially innovative. This means that firms can charge higher royalty rates for licensing interface patents than other patents, regardless of the degree of innovation the interface patents may embody.[61] For these reasons, interface patents are among the most valuable patents that ICT developers can own.

Another incentive to patent interfaces may derive from a perception that other forms of IP protection for interfaces are much weaker than patents. Insofar as outsiders can reverse engineer ICT systems to gain access to interfaces, trade secrets in the interfaces may be vulnerable to appropriation.[62] Determined reverse engineers may be motivated to discover obscure aspects of interfaces.[63] The enforceability of license

---

[58] See, e.g., Atari Games Corp. v. Nintendo of Am., Inc., 30 U.S.P.Q.2d (BNA) 1401 (N.D. Cal. 1993) (developer of complementary product infringed interface patent).

[59] Patents on internal designs of programs are, by contrast, often difficult to enforce offensively (that is, to stop competitors from using them) because such designs are typically difficult to discern from executing commercially distributed object code. Although firms often seek patents for internal design components, such as algorithms and data structures, patents on such innovations are generally more useful for defensive than for offensive purposes. That is, developers tend to seek patents on such internal design elements to assure themselves of having freedom to develop software embodying these inventions as well as to build a portfolio of IP assets so that the firms will have something to trade (e.g., by cross-licensing) if a competitor asserts patent claims against them. See, e.g., Gideon Parchomovsky & R. Polk Wagner, *Patent Portfolios*, 154 U. Pa. L. Rev. 1 (2005)

[60] See, e.g., Bakels & Hugenholtz, supra note xx, at 22 (giving an example). Narrowly drawn interface patents have an advantage over broadly written interface patents because narrow patents are generally easier to defend against invalidity challenges.

[61] See, e.g., Mark A. Lemley, *Ten Things to Do About Patent Holdup of Standards (and One Not to*), 48 B.C. L. Rev. 149 (2007) (discussing holdup problems that arise when firms own patents on standards).

[62] JAMES POOLEY, TRADE SECRET LAW sec. 5.02[1].

[63] Id. at sec. 4.04[4]. See Samuelson & Scotchmer, supra note xx, at 1587, n.49-50.

restrictions on reverse engineering has been widely questioned.[64] Although copyright law protects program code, any interfaces embedded in programs are beyond the scope of copyright's protection.[65] The Ninth Circuit Court of Appeals in *Sega* explicitly suggested that patents may be the only effective way to protect the functional requirements for achieving interoperability.[66] Neither the PTO nor the courts seem to require much disclosure from developers of ICT interfaces.[67] Firms may thus be able to get patents on some aspects of their interfaces while at the same time maintaining detailed specifications of the interfaces as trade secrets. [68]

Established firms are more likely than entrepreneurs to patent their interfaces, as the patents will give them more legal control over those desiring to make compatible systems. Although entrepreneurial firms sometimes launch new platforms for which they have developed novel interfaces, they are more likely to publish interface specifications for their platforms and encourage others to make unrestricted use of the interfaces than to seek patents. [69] Publishing interfaces will enable developers of other software and/or peripheral equipment to produce programs and other products that can interoperate with the new platform.

Insofar as do entrepreneurs seek patents on interfaces,[70] these patents are unlikely to confer substantial market power because entrepreneurs will generally need to license

---

[64] Id. at 1626-30 (reviewing the controversy over enforceability of anti-reverse engineering clauses and why most scholars think such clauses should not be enforced, particularly in mass-market licenses).
[65] See supra notes xx and accompanying text.
[66] *Sega*, 977 F.2d at 1526.
[67] See, e.g., In re Hayes Microcomputer Prods, Inc., 982 F.2d 1527, 1532-39 (Fed. Cir. 1992)(rejecting challenges to interface patent based on inadequacy of written description and best mode disclosure requirements of patent law).
[68] Software developers cannot seek patent protection for documents detailing interface specifications, as such documents would be ineligible for patent protection as "printed matter." [cite] Although copyright protection might be available to an original comprehensive listing of interface details, the scope of this copyright would be very thin, for implementation of the interface in an independently developed program would not infringe copyright in the listings under established caselaw. See, e.g., *Altai*, 982 F.2d at 703; *Sega*, 977 F.2d at 1524.
[69] Entrepreneurs are more likely than not to find themselves in an IP environment in which they will need access to interface information developed by others in order to interoperate with existing platforms and/or applications. If ICT interfaces were excluded from patent protection or owners were required to license them, it would lessen at least one of the IP risks that entrepreneurs face in the marketplace.
[70] U.S. Patent No. 6,125,391, issued in 2000 to Bart Meltzer, et al. is an example of an interface patent issued to entrepreneurs. This patent covered key aspects of Internet business transaction exchanges and was an important asset for a small start-up company, Veo Systems, Inc., that made Veo an attractive acquisition target for Commerce One, which was building an e-commerce platform. Customers of CommerceOne's platform obtained a royalty-free perpetual license to practice the invention. Robert J. Glushko, one of the inventors on the patent, spent much of his time at Commerce One assuring participants in global standard setting processes that Commerce One would grant royalty-free perpetual licenses so that everyone could engage in the interoperable commerce envisioned by the patent. When Commerce One went bankrupt, however, these patents were the most valuable asset that Commerce One owned. A bankruptcy auction brought in $14.5 million for them. Although there was concern within the industry that these patents had been obtained by a "patent troll," it later came out that Novell purchased them and dedicated them to a patent commons. See, e.g., John Markoff, *Auction of Internet Commerce Patents Draws Concern*, N.Y. Times, Nov. 16, 2004; John Markoff, *Secretive Buyer of Some E-Commerce Patents Turns Out to Be Novell*, N.Y. Times, May 5, 2005.

such patents on open or reasonable terms enable others to develop interoperable products. The strongest reason for entrepreneurs to patent interfaces is because such patents may be of interest to venture capitalists and other funding sources for entrepreneurial ICT firms,[71] although some VCs regard software patents as a drag on innovation in the software industry.[72]

It is important to realize that incentives to seek patents for interfaces may change over time. As an entrepreneur's ICT system becomes successful in the marketplace, the entrepreneur may become increasingly proprietary about its interfaces and more inclined to seek patents for extensions of existing interfaces or for new ones.[73] Exerting proprietary control over interfaces is also more likely as growth of the platform flattens, and firms make fewer investments in potentially disruptive innovations and more in maintaining control over the existing market.

D. Examples of Interface Patents

One example of an ICT interface patent that illustrates their potential potency was that which Nintendo got for a relatively high level design for a program-to-program interface for its Nintendo Entertainment System (NES).[74] The NES included a game console, a monitor, and controls to allow users to operate games played on the console. Nintendo also made games for its platform and licensed some firms to do the same. Loaded onto the NES console was an initialization program called 10NES, which served as an authentication protocol so that only games licensed by Nintendo could successfully be played on the NES platform. Nintendo-licensed game cartridges contained a program that interacted with the 10NES program and produced a data stream that, in essence, served as a key to open the 10NES console lock so that games could be played.

Through a combination of reverse and social engineering,[75] Atari Games figured out how to generate a data stream that would allow its games to run on the NES console. After Atari Games began selling these games, Nintendo successfully charged it with infringement of its patent on a system for determining authenticity of an external memory used in an information processing apparatus.[76] By patenting this authentication/interface

---

[71] See, e.g., Ronald Mann, *Do Patents Facilitate Financing in the Software Industry?*, 83 Tex. L. Rev. 961, 972 (2007) (arguing that software patents facilitate financing of startup firms).

[72] See, e.g., Brad Feld, *Abolish Software Patents*, Feld Thoughts, April 10, 2006, available at http://www.feld.com/blog/archives/2006/04/abolish_softwar.html.

[73] Patents on interfaces may, indeed, amplify the network effects noted above. See, e.g., Bakels & Hugenholtz, supra note xx, at 22.

[74] U.S. Patent No. 4,799,635.

[75] The social engineering occurred when Atari Games' lawyer obtained a copy of the 10NES source code by misrepresenting to the Copyright Office the firm's need for the program code for litigation purposes. See Atari Games Corp. v. Nintendo of Am, Inc., 30 U.S.P.Q.2d (BNA) 1401, xx (N.D. Cal. 1993).

[76] Id. Atari Games sought a declaratory judgment of non-infringement as to both copyright and patent infringement claims in response to a threat of litigation by Nintendo. Nintendo counterclaimed for copyright and patent infringement, the former claim based in part on the intermediate copying of Nintendo code in the course of reverse engineering. The Federal Circuit upheld Atari Games' fair use defense as to reverse engineering done for purposes of achieving compatibility with current versions of the 10NES program, but not as to reverse engineering to achieve compatibility with those parts of the 10NES program

system, Nintendo was able to exclude Atari Games from making compatible products for its console and obtain damages for the latter's infringing uses.[77]

A fierce competitor of Nintendo's, Sega Enterprises, licensed a patent on a Trade Mark Security System (TMSS) interface technique which it then embedded in its Genesis video game system in an effort to prevent unlicensed videogame developers from producing games compatible with its popular console.[78] In Sega Enterprises, Ltd. v. Accolade, Inc., the Ninth Circuit Court of Appeals explained:

> When a game cartridge is inserted, the microprocessor contained in the Genesis III searches the game program for four bytes of data consisting of the letters "S-E-G-A" (the "TMSS initialization code"). If the Genesis III finds the TMSS initialization code in the right location, the game is rendered compatible and will operate on the console. In such case, the TMSS initialization code then prompts a visual display for approximately three seconds which reads "PRODUCED BY OR UNDER LICENSE FROM SEGA ENTERPRISES LTD."[79]

The *Sega* decision is mainly known for its ruling that Accolade made fair use of Sega's copyrighted programs when it disassembled them to discern information necessary to make Genesis-compatible games.[80] However, Sega also sued Accolade for trademark infringement because the Sega trademark popped up when Accolade's games were played on the Genesis console. Because TMSS was essential to achieving interoperability with the Sega platform, the Ninth Circuit ruled there was no trademark infringement.[81] Had Sega owned the patent on TMSS, it probably could have stopped Accolade from making games for its platform.

Patents on communications protocols have had powerful exclusionary effects in two other litigated cases involving widely used ICTs. One was a patent on an improved method for controlling modes of modem operations that Hayes Microcomputer embodied in its SmartModem products which became a de facto standard in the modem market.[82] Not only did software developers have to implement this protocol when developing software to interoperate with Hayes' modems, but so did rival producers of modems. Modems are used to modulate and demodulate signals, both analog and digital, that

---

that might be used to thwart compatibility in the future. See Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 839-40 (Fed. Cir. 1992). The lower court subsequently granted summary judgment to Nintendo on patent claims. See *Atari Games*, 30 U.S.P.Q.2d at 1414.

[77] The court concluded that Atari Games was a contributory infringer, not a direct infringer, of this patent. Id. Both the patent and copyright claims in this case are discussed at length in Julie E. Cohen, *Reverse Engineering and the Rise of Electronic Vigilantism: Intellectual Property Implications of "Lock-Out" Programs*, 68 S. Cal. L. Rev. 1091 (1995).

[78] See U.S. Patent No. 4,462,076 (videogame cartridge recognition and security system).

[79] 977 F.2d 1510, 1527 (9th Cir. 1992).

[80] Id. at 1520-27.

[81] Id. at 1528-30. Sega was a licensee of the TMSS patent, not its owner, so Sega did not bring a patent infringement suit against Accolade for the latter's use of the TMSS. Id. at 1524, n. 7.

[82] U.S. Patent No. 4,549,302. The patent and some of its claims are discussed in In re Hayes Microcomputer Prods, Inc., 982 F.2d 1527 (Fed. Cir. 1992).

enable communications between telephones and computers. Modems have two modes: a transparent mode in which the modem performs its modulation-demodulation functions, and a command mode in which modems respond to predetermined commands and perform operations by executing instructions in firmware.[83] The predetermined command, which Hayes arbitrarily designated as "+++_", instructed the modem when to switch between transparent and command modes. Ven-Tel was one of 125 modem manufacturers whose modems were compatible with this feature of Hayes' modems. Although Ven-Tel challenged the validity of this patent, a jury upheld it and found infringement; the Federal Circuit affirmed.[84]

A more recent example of patent infringement involving a commercially significant interface design was a lawsuit brought by Verizon Services Co. against Vonage America, Inc.[85] Vonage began providing Voice over Internet Protocol (VoIP) telephone service to customers in 2002; by the time Verizon sued it for patent infringement, Vonage had 2.2 million customers. Verizon's patents covered methods of enhanced translation of telephone numbers into and from Internet Protocol addresses, which facilitated more effective interconnection of VoIP services with telephone network services. A jury ruled against Vonage's challenge to the patent's validity and awarded Verizon $58 million in damages.[86] The trial judge stayed injunctive relief, pending Vonage's appeal. The Federal Circuit affirmed the finding of infringement as to two of the patents, although remanding the case for reassessment of damages; yet, it affirmed the issuance of an injunction.[87] Within the Internet telephony community, concerns arose about the implications of Verizon's patents for VoIP services more generally.[88]

Although Microsoft has not brought suit for infringement of interface patents, it has sought and obtained a substantial number of patents on protocols for its computer programs in recent years. It holds, for example, 65 U.S. patents and 6 European patents on work group server and program protocols,[89] and numerous other patent applications for similar protocols are pending.[90] Microsoft relied on some of these patents as a justification for refusing to provide and license interface information to Sun

---

[83] Id. at 1531.
[84] Id. at 1530.
[85] Verizon Services Co. v. Vonage America, Inc., 503 F.3d 1295 (Fed. Cir. 2007).
[86] Id. at 1301-02.
[87] Id. at 1311. Vonage relied upon the Supreme Court's decision in eBay, Inc. v. MercExchange L.L.C., 547 U.S. 388 (2006) (holding that courts have discretion not to issue injunctions in patent infringement cases) in support of its argument that the public interest would be served by an award of damages in lieu of an injunction. Id. at 1310-11. The Federal Circuit was not persuaded. Id. The implications of the *eBay* case for interface patents is discussed infra notes xx and accompanying text. Vonage was able to make an arrangement with a VoIP network services provider to carry calls placed by Vonage's customers. See Eric Bangeman, *Vonage Hangs Up on Verizon Patent Infringement with New Agreement*, Ars Technica, April 2, 2007, available at http://www.arstechnica.com/news.ars/post/20070402-vonage-hangs-up-on-verizon-patent-infringement.html
[88] See, e.g., Adario Strange, *The Future of Internet Telephony Could Hang on the Vonage Case*, WIRED, April 26, 2007, available at http//www.wired.com/print/techbiz/it/news/2007/04/vonage_appeal.
[89] Microsoft Corp., WSPP Patent Mapping, Feb. 1, 2008, available at http://download.microsoft.com/download/2/8/a/28a250e5-5b79-4547-9959-346736ed7a97/WSPP_Patent_Mapping.pdf (hereafter "Microsoft Patent Mapping").
[90] Id.

Microsystems and others in a case brought by the European Commission charging it with abuse of dominant position.[91] Microsoft also owns an interface patent on aspects of its Advanced Streaming Format (ASF).[92] Some open source programmers have wanted to write import/export filters for ASF. Because doing so would infringe Microsoft's patent, this follow-on software product has not been developed.[93]

These examples show that established firms with strong market positions and/or market power sometimes seek and obtain patents on interfaces that increase their ability to control the development of competing and complementary products. The next section will consider various policy responses that have been identified for dealing with the exclusionary power of such patents.

II. Policy Options for Responding to Interface Patents

This Part will discuss the remarkably varied array of policy options that have been proffered in the past two decades as possible responses to the potency of patents on ICT interfaces. Although owners of patents on interfaces would likely argue that they should be able to exercise their patent rights however they choose,[94] most commentators, policymakers, and courts have favored some regulation of interface patents.

Section A reviews proposals to exclude interfaces from patent protection or to immunize use of patents insofar as this is essential for interoperability. Section B considers proposals to adapt or reform patent laws to facilitate interoperability. Section C discusses proposals to use liability rules rather than property rules as to unauthorized uses of interface patents. Section D assesses the role of antitrust and competition law in regulating refusals to license patents on ICT interfaces and/or to supply information necessary to achieving interoperability. Section E sets forth several private sector initiatives for dealing with patents on interfaces essential for interoperability.

A. Banning Patents on Interfaces or Immunizing Their Use

ICT interfaces may be so essential to achieving interoperability that some believe this justifies excluding interfaces from the realm of patentable subject matter. Sun Microsystems, for instance, has taken this position in some public policy debates.[95] Some Sun executives believe that interfaces affecting interoperability should be free from

---

[91] Microsoft's IPR defense in this case is discussed at length infra Part II-E.

[92] Foundation for a Free Information Infrastructure, *Software Patents in Action*, available at http://eupat.ffii.org/patents/effects/index.en.html (compilation of news stories and case studies illustrating the impacts of software patents on the software industry, particularly as to open source software development).

[93] Id. at 2.

[94] I have yet to find a single article or policy document that endorses the view that owners of patents on interfaces should be able to exercise these patents as they wish. The Federal Circuit's decisions in the *Hayes* and *Verizon* cases (see supra notes xx and accompanying text) are, however, consistent with this position, as are briefs submitted by Microsoft in the European Commission's competition proceeding discussed in Part II-E.

[95] Band & Katoh, supra note xx, at 332-34.

IP restrictions and be treated as a commons on which all comers are free to build.[96] Alternatively, some have proposed abolishing software patents altogether, which would obviously sweep away patents on computer program interfaces.[97]

The European Commission's recent interpretation of the 1991 directive on the legal protection of computer programs may provide support for the abolish-interface-patents movement.[98] In its proceeding against Microsoft for abuse of dominant position based on that firm's refusal to supply interface information to Sun and others and license its use, the Commission flatly denied that Microsoft owned any IPRs in the interfaces the Commission ordered it to disclose to Sun and others.[99] The interfaces, in the Commission's view, were ideas and principles under the Software Directive.[100]

Although the EU Software Directive is often viewed as having endorsed copyright as a means of legal protection for computer programs,[101] the Directive is better understood as having created a sui generis (of its own kind) form of protection for computer programs, especially as regards interfaces and interoperability, under the guise of copyright.[102] The Software Directive gives computer programs an unusually "thick"

---

[96] Conversation with Greg Papadopolos, Chief Technology Officer and Executive Vice President for Research and Development, Sun Microsystems, May 8, 2008, San Jose CA. This theory builds on the work of Yochai Benkler and Jonathan Zittrain who perceive the open and commons-like infrastructure of the Internet as having enabled innovation, competition, and other socially desirable results. See, e.g., YOCHAI BENKLER, THE WEALTH OF NETWORKS (2006); JONATHAN ZITTRAIN, THE FUTURE OF THE INTERNET—AND HOW TO STOP IT 78-79(2008). See also Brett M. Frischmann, *An Economic Theory of Infrastructure and Commons Management*, 89 Minn. L. Rev. 917, 918-19 (2005). Some private initiatives aimed at muting the exclusionary power of interface patents, discussed infra Part II-C, are consistent with the commons approach toward interfaces essential to interoperability.

[97] A coalition of nonprofit organizations affiliated with the free and open source software movements has formed to espouse the abolition of software patents. See http://endsoftpatents.org. At least one venture capitalist agrees with the gist of this coalition's arguments. See, e.g., Brad Feld, *Abolish Software Patents*, Feld Thoughts, April 10, 2006, available at http://www.feld.com/blog/archives/2006/04/abolish_softwar.html. There is reason to question the patentability of software innovations as a matter of U.S. law. See, e.g., Pamela Samuelson, Benson *Revisited: The Case Against Patent Protection for Algorithms and Other Program-Related Inventions*, 39 Emory L.J. 1025 (1990) (discussing various arguments against patents on computer program and other information innovations). See also infra notes xx and accompanying text.

[98] Software Directive, supra note 3. Although the Commission's initial order against Microsoft indicated that the Commission did not have enough information about Microsoft's claimed IPRs to make a judgment about the extent of these rights, see EC Decision, supra note 4, at par. 190, the Commission defended its order before the CFI by asserting that Microsoft had *no* IPRs in its interfaces, see CFI Decision, supra note 4, par. 276-78. It is, of course, possible that the Commission was merely questioning the validity of Microsoft's interface patents. See id. at 276. But the CFI, in discussing the Commission's assertions, mentioned the Software Directive's recitals that refer to interfaces as ideas and principles. As I reflected on the Commission's denial that Microsoft had IPRs in its interfaces and the reference to the Directive's recitals, I generated the sui generis interpretation of the Directive discussed in this section.

[99] Id.

[100] Id.

[101] Software Directive, supra note 3, Art. 1.1.

[102] I have previously argued in favor of a sui generis form of legal protection for programs. See, e.g., Pamela Samuelson, et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 Colum. L. Rev. 2308 (1994). Within the framework set forth in the Manifesto, program interfaces would be considered industrial compilations of applied know-how which would be eligible for a short period of

scope of protection as to most of the underlying structure of programs.[103] However, the Directive defines interfaces necessary for interoperability as unprotectable ideas and principles,[104] even though they may be very important and commercially significant elements of program structure.

The decompilation provisions of the Directive, which are also sui generis parts of its framework, reinforce the thick protection for most internal program structure because it is illegal under the Directive to decompile a program, which necessarily involves making copies of the program, to get access to its internal designs, unless the decompiler is trying to get access to interface information.[105] In essence, the Council made copyright law into a super-strong trade secrecy law as to every aspect of program internals—except interfaces.

Under the Directive, published interfaces, as ideas and principles, are in the public domain and available for free copying. Embedded in program code, interfaces remain unprotected ideas and principles, although they can be hidden away if the program's developer distributes its code in machine-executable form, as is common in the software industry.[106] The Directive contemplates that those who want to develop interoperable programs can gain lawful access to these secrets in one of two ways: either by licensing the interface information from the software's developer or by reverse engineering the code to extract interface information. The latter option is, however, only available under the Directive if the information is not readily available on reasonable terms from the program's developer.[107] This suggests that the Directive's intent is not to encourage reverse engineering activities, but rather to induce firms to license interface information

---

exclusivity, following which others could use the interfaces, subject to an obligation to compensate the interface's developer. While it is still accurate to characterize interfaces as industrial compilations of applied know-how, I acknowledge that the Commission's approach of treating interfaces as unprotectible ideas or principles avoids a problem that the Manifesto did not address, namely, the likelihood that firms whose blocking periods for interfaces were about to expire would have incentives to revise them to ensure continued exclusivity for them and thereby thwart compatibility with unlicensed parties' products.

[103] See generally BRIDGET CZARNOTA & ROBERT J. HART, LEGAL PROTECTION OF COMPUTER PROGRAMS IN EUROPE—A GUIDE TO THE EC DIRECTIVE (1991). For a comparison of U.S. and EU law in respect of protection of internal structure of computer programs, see Pamela Samuelson, *Comparing U.S. and E.C. Copyright Protection For Computer Programs: Are They More Different Than They Seem?*, 13 J. Law & Comm. 279 (1994).

[104] Software Directive, supra note 3, Art. 1.2. See also id., Recitals 10-13.

[105] Id., Art. 6.1.

[106] In addition to denying that Microsoft had any patents in its WGS-OS interfaces, the European Commission rejected Microsoft's claim that it had protectable trade secret interests in the detailed interface information it wished to withhold from Sun and other competitors. See CFI Decision, supra note 4, at par. xx. The logic of the Commission's position flows from the Directive having deemed interface information to be unprotectable ideas and principles, its having authorized reverse engineering to get access to these ideas and principles, and its giving very strong IP protection to other program internals other than interfaces, such that the Commission thought trade secrecy would not be necessary for program details.

[107] Software Directive, supra note 3, Art. 6.1(b). The Software Directive also makes clear that it is not lawful to reverse engineer any parts of the program other than those that contain interface information. Id.

on reasonable terms because if they don't, would-be interoperators will be able to lawfully reverse engineer the code to extract the information.[108]

Another sui generis provision of the Directive protects interfaces from a market-destructive loss of secrecy by limiting what lawful reverse engineers can do with information about interfaces that they extract through reverse engineering. Engineers are authorized to use the information to develop an independently developed program that interoperates with the reverse-engineered program, but they are forbidden from disclosing the reverse-engineered interface information to others.[109] Each firm that wants to develop an interoperable program must thus undertake the same tedious reverse engineering process to get access to interface information if it is unable to license the information from the first program's developer. To ensure that the inducement to licensing is not thwarted, the Directive also provides that the decompilation privilege cannot be contracted away.[110]

If, as I suggest here, the Software Directive adopted a sui generis rule against IP protection for interfaces in the EU, it would follow that patent protection should not be available for interfaces either.[111] European patent law, like U.S. patent law, does not allow ideas or principles to be protected by patents.[112] Characterizing interfaces necessary to interoperability as ideas and principles would logically mean that the interfaces should be regarded unprotectable under the Directive.[113] The Commission may have had this interpretation in mind when it denied that Microsoft had valid patents on its interfaces.[114]

---

[108] See, e.g., Samuelson & Scotchmer, supra note xx, at xx. Induced licensing has the advantage of getting some compensation to the developer of the interface, while at the same time ensuring that second comers have the information they need to make their systems interoperable.

[109] Software Directive, supra note 3, Art. 6.2.

[110] Id., Art. 9.1.

[111] Whatever the merits of my discussion of the Directive as a sui generis regime as to interfaces, it is clear that the European Patent Office does not interpret the Software Directive as precluding patents on program interfaces, for it has issued patents on interface designs to Microsoft. See Microsoft Patent Mapping, supra note xx.

[112] See, e.g., Bakels & Hugenholtz, supra note xx, at 28. See also European Patent Convention, Art. 52.

[113] Software Directive, supra note 4, at Art. 9.1. Other functional design elements of programs, apart from interfaces, may be eligible for patents, although there is less need for firms to patent program structure in the EU, since the Directive provides such a thick scope of protection for program internals, which can be had without concomitant disclosure requirements. Bakels & Hugenholtz, supra note xx, at xx. See also Samuelson, supra note xx, at xx (explaining why the EU provides a broader scope of protection to software than US law does).

[114] CFI Decision, supra note 4, par. 278. Reinforcing this interpretation of the Software Directive is a provision in the proposed European Software Patent Directive, supra note xx, that defined the relationship contemplated between it and the 1991 directive: "The rights conferred by patents granted for inventions within the scope of this [software patent] directive shall not affect acts permitted under Articles 5 and 6 of Directive 91/250/EEC on the legal protection of computer programs by copyright, in particular under the provisions thereof in respect of decompilation and interoperability." See Robert Bray, *The European Union "Software Patents" Directive: What Is It? Why Is It? Where Are We Now?*, 11 Duke L. & Techn. Rev. (2005), pars. 28 (setting forth this provision). If it was lawful to reverse engineer a program to get access to interface information under the 1991 directive and to use that information to develop an interoperable program, this provision suggests those acts would still be lawful after adoption of the software patent directive. By implication, reuse of interfaces could not be blocked by patents because

The Court of First Instance (CFI) decided it was unnecessary to resolve whether Microsoft had any IP rights in its interfaces, although it assumed for the sake of its review of the Commission's order that Microsoft did.[115] The issue of whether the Software Directive excludes interfaces from patent protection has thus been left for another day.

Program interfaces are much likely to be regarded as patentable subject matter under U.S. law under the very broad conception of patentable subject matter articulated in State Street Bank & Trust Co. v. Signature Financial Group, Inc.[116] That decision considers everything under the sun made by humans to be patentable subject matter as long as it produces a useful concrete and tangible result.[117] Program interfaces, as human-made designs that result in information being exchanged across ICT system boundaries, would seem to produce a useful concrete and tangible result.

The *State Street* conception of patentable subject matter may, however, be unsound. Several members of the U.S. Supreme Court have recently questioned the Federal Circuit's capacious view of patent subject matter,[118] which is plainly inconsistent with prior rulings of the Court.[119] Seemingly emboldened by the higher court's questions, the U.S. Patent & Trademark Office (PTO) has started rejecting applications for claiming unpatentable subject matter.[120] The Federal Circuit has distanced itself somewhat from *State Street* by affirming PTO rejections of two claims on subject matter

---

interfaces are ideas and principles. Although the European Parliament ultimately rejected the Proposed Software Patent Directive, it remains to be seen whether the courts will interpret the Software Directive as precluding patent as well as copyright protection for program interfaces.

[115] CFI Decision, supra note 4, at par. 283.

[116] 149 F.3d 1368 (Fed. Cir. 1998). See also AT&T Corp. v. Excel Comm'ns, Inc., 172 F.3d 1352 (Fed. Cir. 1999)(accord as to expansive conception of patent subject matter).

[117] *State Street*, 149 F.3d at 1373.

[118] The Supreme Court granted certiorari in LabCorp. v. Metabolite Labs., 543 U.S. 1185 (2005) to review whether a method of correlating information about the amount of a certain chemical in a patient's bloodstream and diagnosing that the patient had an abnormal condition claimed patentable subject matter. The Court ultimately decided that the writ had been improvidently granted, apparently because the subject matter issue had not been cleanly presented below. However, Justice Breyer wrote a powerful dissent, joined by two other Justices, that called into question the Federal Circuit's patentable subject matter standard. See LabCorp. v. Metabolite Labs, 126 S.Ct. 2621, 548 U.S. – (2006). (Information about this case and briefs filed before the Court are available at http://patentlaw.typepad.com/patent/2006/01/supreme_court_l.html.) Several Justices also asked questions about patentable subject matter during the oral argument in another recent patent case before it. See Pamela Samuelson, *Software Patents and the Metaphysics of 271(f)*, 50 Comm. ACM 15 (June 2007).

[119] See, e.g., Brief Amicus Curiae for Consumers Union, Electronic Frontier Foundation, and Public Knowledge at 22-24, submitted to the Court of Appeals for the Federal Circuit in In re Bilski, April 7, 2008, available at http://www.eff.org/files/CU-EFF-PK-Bilski-Amicus.pdf (asserting that *State Street* is inconsistent with the Court's patentable subject matter jurisprudence).

[120] Since 2006, the PTO has rejected numerous claims for failure to claim patentable subject matter. See, e.g., Ex Parte Yang-Huffman, 2007 WL 2899992 (PTO Bd. Pat. App. & Interf. 2007)(rejecting claim for method for dynamic configuration of information); Ex Parte Gosby, 2007 WL 2843739 (PTO Bd. Pat. App. & Interf. 2007)(rejecting claim for method of document analysis and retrieval); Ex Parte Gutta, 2007 WL 1766997 (PTO Bd. Pat. App. & Interf. 2007)(rejecting claim for method for evaluating closeness of two items).

grounds,[121] and by rehearing a third such case en banc in May 2008.[122] The Federal Circuit invited briefs to consider whether the method at issue claimed patentable subject matter as well as whether the *State Street* subject matter ruling should be refined or repudiated.[123]

It is too early to know whether these decisions will make interface patents vulnerable to challenge on subject matter grounds, but it is possible insofar as they are for methods of representing data, methods of calculating numbers, or methods of information exchange.[124] The Federal Circuit is very unlikely to rule that software innovations are per se unpatentable,[125] and even the Supreme Court may not go that far. Consequently, interface patents, insofar as they are for technological processes, will probably be no more vulnerable to subject matter challenges than other technical innovations. U.S. courts are also unlikely to be swayed by policy-based arguments against patents for interfaces essential to interoperability.[126]

Congress could, of course, legislate an exclusion of interfaces from patent protection.[127] But at this point, there is insufficient momentum or consensus in the U.S. policy arena about the importance of interoperability and patents as an impediment to interoperability to make it likely that Congress would consider excluding patents for interface innovations in order to achieve it.[128]

An alternative strategy for restricting interface patents is to allow them to issue, but deem their use non-infringing if essential for achieving interoperability. During the period when the European Parliament was considering whether to adopt a directive on the patenting of software-related inventions,[129] there was a proposal that use of any patents

---

[121] In re Comiskey, 499 F.3d 1365 (Fed. Cir. 2007) (method for conducting arbitrations through the use of legal documents held unpatentable process); In re Nuijten, 500 F.3d 1346 (Fed. Cir. 2007) (encoded signal held unpatentable subject matter).

[122] Ex Parte Bilski, 2006 WL xx (PTO Bd. Pat. App. & Interf. 2006).

[123] Per Curiam Order, In re Bilski, U.S. Court of Appeals for the Federal Circuit, No. 2007-1130 Feb. 15, 2008.

[124] See, e.g., *Benson* Revisited, supra note xx, at xx.

[125] The Court might decide that program code is unpatentable subject matter, see PTO Guidelines (programs as such excluded from patent protection), and might reaffirm the unpatentability of broad abstract algorithms, as in *Benson*. However, functional designs and processes embedded in software seem likely to remain patentable subject matter.

[126] See, e.g., Verizon Services Corp. v. Vonage Holdings, 503 F.3d 1295, 1310 (Fed. Cir. 2007) (affiming issuance of an injunction for infringement of patent on interconnection technique against the leading Voice Over Internet Protocol (VOIP) telecommunications service, rejecting argument that damages would suffice to protect Verizon's interests).

[127] Congress is currently considering legislation that would exclude tax planning methods from patent protection. See H.R. 1908, Patent Reform Act of 2007, 110th Cong., 1st Sess., sec. 10 (2007); S. 2369, 110th Cong., 1st Sess. (2007). See also Gerald J. Mossinghoff, *Remedies Under Patents on Medical and Surgical Procedures*, 78 J. Pat. Tm. Off. Soc'y 789 (1996) (discussing proposals to exclude medical and surgical procedures from patent subject matter).

[128] In the free and open source software community, there is a strong concern about interface and other software patents as threatening to the viability of this sector of the software industry. See, e.g., Amy Kucharik, *Lingering Patent Threats Worry Open Source Experts*, Linuxworld, Feb. 16, 2005; FFII, supra note xx (giving examples of software patents that have impeded interoperability).

[129] See Proposed Software Patent Directive, supra note xx.

that "read" on interfaces should be deemed non-infringing insofar as there was no equally efficient or effective alternative non-patented way to achieve interoperability. [130] The European Parliament did not adopt a software patent directive, so this provision was not adopted.[131]

While no similar legislative proposal has been introduced in the U.S. Congress, there is legislative precedent for immunizing socially productive uses of patented techniques.[132] Section 287(c) of U.S. patent law immunizes doctors from liability for using patented medical or surgical procedures to treat patients.[133] If a strong social consensus developed in favor of interoperability, Congress might well adopt a similar rule for immunizing use of patents as to interfaces essential to interoperability.[134]

Finally, Professor Julie Cohen has argued that owners of patents on interfaces should be deemed to have misused patents insofar as the patents are used as lock-out devices.[135] Cohen used Nintendo's success in asserting a patented authentication method to stop Atari Games from making and selling games that could run on the Nintendo platform as an example of this. Cohen contended that Nintendo's exercise of this patent unlawfully extended the patent's scope, in essence, creating an unlawful tying arrangement between the Nintendo console and Nintendo-licensed games.[136] The patent covered only the authentication technique, not the games or consoles; yet, Nintendo was able to exercise the patent to control the making and selling of games for the platform,

---

[130] The proposed Article 6a would have provided: "Member States shall ensure that wherever the use of a patented technique is needed for the sole purpose of 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 patent infringement." See Bray, supra note xx, at parag. 22.

[131] Foundation for a Free Information Infrastructure, European Parliament Says No to Software Patents, July 6, 2005, available at http://wiki.ffii.org/Ep050706En.

[132] See H.R. 2365, 110th Cong., 1st Sess. (2007), under which taxpayers, tax practitioners, and related professional organizations would be immune from liability for use of any patent on a tax planning method. See also H.R. 5638, 110th Cong., 1st Sess. (2007) which proposes immunity from patent infringement liability for firms supplying repair parts.

[133] 35 U.S.C. sec. 287(c). This provision was adopted after the American Medical Association and several other leading organizations of physicians lobbied for immunity after a surgeon was sued for infringing another surgeon's patented technique for cataract surgery. Mossinghoff, supra note xx, at 795-97. Mossinghoff believes this immunity provision is compatible with U.S. obligations under the World Trade Organization Agreement on Trade Related Intellectual Property Rights (TRIPs). Article 30 of TRIPs allows WTO members to create "limited exceptions to the exclusive rights conferred by a patent provided that such exceptions do not unreasonably conflict with a normal exploitation of the patent and do not unreasonably prejudice the legitimate interests of the patent owner, taking account of the legitimate interests of third parties." Because the overwhelming majority of patents in respect of medical procedures are owned by biotechnology and pharmaceutical firms, none of whom sues doctors for treating patients, Mossinghoff argues that sec. 287(c) does not conflict with the normal exploitation of these patents. But see Emily C. Melvin, Note, *An Unacceptable Exception: The Ramifications of Physician Immunity from Medical Procedure Patent Infringement Liability*, 91 Minn. L. Rev. 1089 (2007) (challenging both the wisdom of sec. 287(c) and its compatibility with TRIPS).

[134] Justifying an interoperability exception to patent enforcement under the TRIPS Agreement may be more difficult because a normal exploitation of interface patents may include licensing them.

[135] Cohen, supra note xx, at 1182-83.

[136] Id.

even though the games were not within the scope of the patent.[137] The patent on this small but crucial component of Nintendo's ICT system conferred power over many innovations that were well beyond the patent's scope.

B. Tailoring Patent Rules Affecting Interoperability and Patent Reforms

There are several ways that patent rules can be tailored to lessen the risks that patents on interface techniques will impede interoperability. Professor Cohen has suggested one such way, namely, heightening the standard of nonobviousness for ICT interface patent claims.[138] Cohen worries that firms will seek patents for interface designs for anti-competitive purposes, that is, as a tool for blocking competitors from developing compatible platforms (e.g., game consoles) and for controlling the market for complementary products (e.g., videogames that run on the patentee's platform).[139] To ensure that patents are being issued only to truly inventive interfaces, Cohen would have the PTO apply an "innovative programmer" standard to judging patentability.[140] Under this standard, she thinks fewer interface patents will issue.

Although patents on ICT interfaces have not (yet) been subjected to a higher non-obviousness standard, patent examiners sometimes do scrutinize some patent applications more closely than others. Business method patent applications, for instance, are reviewed by a "second set of eyes" as a precaution against issuing patents on obvious business methods or on overbroad claims.[141] Professors Burk and Lemley have shown that there are many policy levers in patent law that can be used to respond to industry-specific challenges.[142] The PTO probably does have inherent authority to scrutinize interface patents more carefully than others. If the PTO came to perceive interface patents as potentially being sought for anti-competitive purposes, that might well justify a closer look. However, I do not think the PTO is currently doing this or that the Federal Circuit would condone the practice if it was challenged.

A second, if more indirect, way to tailor patent rules to facilitate interoperability would be to treat the act of reverse engineering an ICT interface for the purpose of getting access to interface information as non-infringing of patents for interfaces or other functional designs embodied in the system.[143] Professor Maureen O'Rourke has proposed that patent law should have a "fair use" defense to enable reuses of patented

---

[137] Misuse of an interface patent would render it unenforceable against those who bypassed the lockout system, such as AG. So under Cohen's proposal, Nintendo would not have been able to enjoin AG, even if it literally infringed the patent.

[138] Id. at 1152-81.

[139] Id. at 1152-53.

[140] Id. Cohen thinks that the 1ONES patent would have been invalid under this heightened standard. Id. at 1153, 1162.

[141] See, e.g., Emerson H. Tiller, *The Business Method Patent Myth*, 18 Berkeley Tech. L.J. 987, 994-95 (2003) (discussing legislative proposals to raise the level of scrutiny of business method patents). Tiller mentions that H.R. 5364, The Business Method Patent Improvement Act, 106th Cong., 2d Sess. (2000) initially called for heightening the nonobviousness standard for business method patents. Id.

[142] Burk & Lemley, supra note xx.

[143] It is, of course, a separate question whether reimplementing the interface or other patented design in a follow-on product would infringe the patent.

software-related innovations to facilitate interoperability.[144] Professors Cohen and Lemley have made a similar recommendation to allow reverse engineering of program code insofar as the reverse engineering was intended to enable the development of interoperable programs, even if it was necessary to use a patented invention in the course of reverse engineering.[145] Cohen and Lemley invoke a number of patent doctrines as possible bases for this policy,[146] but they agree with O'Rourke that the social interest in promoting interoperability should be recognized as important in patent law as well as in copyright and trade secrecy law.[147]

Another doctrinal development that would facilitate greater interoperability would be judicial refusal to enforce anti-reverse engineering clauses of license agreements insofar as these agreements conflict with public policies favoring interoperability in intellectual property law.[148]

In theory, reverse engineering should be unnecessary insofar as the patents cover interface components, as the patent should reveal information necessary for someone skilled in the art to make the patented invention. However, it is well-known that patents for software innovations disclose relatively little, in general.[149] Firms that patent interfaces may well claim the interface technique at a sufficiently high level of abstraction that an ordinary programmer would not be able to create interoperable components.

To respond to this concern, patent examiners could be instructed to be more rigorous about the adequacy of disclosure as to interface techniques. Yet, for already issued interface patents, some reverse engineering may be necessary to extract interface information. Such reverse engineering, say O'Rourke, Cohen & Lemley, should not infringe the interface patent.

Other patent reform measures can also be significant in ensuring that patents on interface designs do not unduly interfere with interoperability. Especially useful would be implementation of two important patent reforms recommended by the Federal Trade Commission and National Academies of Sciences: (1) a reinvigoration of the non-

---

[144] See Maureen O'Rourke, *Toward a Doctrine of Fair Use in Patent Law*, 100 Colum. L. Rev. 1177 (2000).

[145] See Cohen & Lemley, supra note xx, at xx.

[146] Id. at 29-37 (exploring limitations on patent protection for experimental uses, implied license, and exhaustion of rights doctrines as possible bases for interoperability-based defenses to patent infringement claims).

[147] Id. at 27-28. Cohen & Lemley also recommend a narrow scope for the doctrine of equivalents in software-related patent cases, in large part of widespread and legitimate concerns about the low quality of patents in this field. Id., Part III.

[148] See, e.g., Samuelson & Scotchmer, supra note xx, at xx; Mark I. Koffsky, *Patent Preemption of Computer Software Contracts Restricting Reverse Engineering: The Last Stand?*, 95 Colum. L. Rev. 1160 (1995); David A. Rice, *Public Goods, Private Contract and Public Policy: Federal Preemption of Software License Prohibitions Against Reverse Engineering*, 53 U. Pitt. L. Rev. 543 (1992). But see Bowers v. BayState Technologies, Inc., 320 F.3d 1317 (Fed. Cir. 2003) (affirming enforcement of anti-reverse engineering clause in software shrinkwrap license).

[149] See, e.g., JAMES BESSEN & MICHAEL MEURER, PATENT FAILURE, Chaps. 7, 9 (2008).

obviousness standard for attaining patent protection,[150] and (2) a more cost-effective way to challenge invalid patents than the litigation and re-examination procedures under current patent law.[151]

These two reforms are interlinked because when the non-obviousness standard is too low, some patents will have issued that should not have; yet the costs of litigation are so high, that some invalid patents may not be challenged that should be. In KSR Int'l, Co. v. Teleflex, Inc.,[152] the Supreme Court overturned the Federal Circuit's non-obviousness standard as insufficiently rigorous. Unfortunately, many patents were issued under the earlier standard. There is thus an urgent need for a cost-effective post-grant review system to allow those who have strong arguments about patent invalidity to pursue them without the need to resort to lengthy and costly patent litigation. Post-grant review has been a key component of the patent reform legislation that has been pending before Congress in recent years.[153]

While reinvigorating the non-obviousness standard and an improved post-grant review process are reforms that are not specifically aimed at patents on interfaces, there is reason to think that these reforms would be particularly useful to challenge "bad" interface patents. As noted in Part I, firms have incentives to seek patents for interfaces, even when they embody trivial or arbitrary differences from the prior art in order for the firm to have strong exclusionary rights against others. Apportionment of damages based on the value of the technical contribution that the patented invention might also have implications for lowering the risk of substantial liability as to minimally innovative interface techniques.[154]

C. Employing Liability Rules for Use of Interface Patents

Some commentators and policymakers have called for a liability rule approach to patents on interfaces.[155] They would allow unlicensed persons to implement patented interfaces to achieve interoperability as long as the unlicensed persons offer reasonable

---

[150] FEDERAL TRADE COMMISSION, TO PROMOTE INNOVATION: THE PROPER BALANCE OF COMPETITION AND PATENT LAW AND POLICY 10-12 (Oct. 2003) (hereafter "FTC Report").

[151] Id. at 7-8.

[152] KSR Int'l Co. v. Teleflex, Inc., 127 S.Ct. 1727, 550 U.S. – (2007). Reinvigorating the nonobviousness standard was a patent reform that would have been difficult to achieve in Congress because some patent lawyers and industry groups have profoundly different views on how rigorous the nonobviousness standard should be. The Supreme Court's *KSR* decision seems to have achieved this reform in patent law; there has been no effort to reverse the Court's decision through further legislation. See, e.g., Pamela Samuelson, *Patent Reform Through the Courts*, 50 Comm. ACM 17 (Feb. 2007).

[153] H.R. 2365, supra note xx.

[154] The IT industry has strongly supported apportionment of damages in patent cases so that "only [the] economic value properly attributable to the patent's specific contribution over the prior art" would be awarded. See H.R. 1908, supra note xx. This proposal has, however, proven to be controversial. See, e.g., Dennis Crouch, *Patent Reform 2007: Apportionment of Damages*, available at http://www.patentlyo.com/patent/2007/05/patent_reform_2_1.html.

[155] For a general discussion of liability v. property rules, see e.g., J.H. Reichman, *Of Green Tulips and Legal Kudzu: Repackaging Rights in Subpatentable Innovation*, 53 Vand. L. Rev. 53 1743 (2000); Guido Calabresi & Douglas Melamed, *Property Rules, Liability Rules and Inalienability: One View of the Cathedral*, 85 Harv. L. Rev. 1089 (1972).

compensation to the patentee. A liability rule approach can be implemented in a number of ways.

Professor Peter Lee has proposed that courts should withhold injunctive relief for infringement of patents on interfaces essential for interoperability.[156] Lee draws upon the Supreme Court's ruling in eBay, Inc. v. MercExchange, L.L.C.[157] which rejected the Federal Circuit's rigid rule that courts must virtually always issue injunctions in patent cases.[158] The Court in *eBay* observed that under traditional principles of equity, a plaintiff is not entitled to issuance of an injunction unless it shows that (1) it has suffered irreparable injury, (2) remedies at law are inadequate to compensate it for the injury, (3) a remedy in equity is warranted in view of the balance of hardships between the plaintiff and defendant, and (4) the public interest would not be disserved thereby.[159]

Especially relevant to Lee's argument is Justice Kennedy's concurrence in *eBay*, which was joined by three other Justices.[160] It recognizes that some firms nowadays use patents "as a bargaining tool to charge exorbitant fees to companies that seek to buy licenses to practice the invention."[161] This problem is especially acute "when 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."[162] In such cases, "legal damages may well be sufficient to compensate for the infringement and an injunction may not serve the public interest."[163]

In its amicus curiae brief to the Court in *eBay*, Nokia Corp. focused on the risk that patents on interfaces could impede socially beneficial interoperability.[164] The Federal Circuit's rigid rule on injunctions could, it said, "particularly encumber the technologically sophisticated industries that fuel the national economy's growth" because these industries rely on "interoperability standards—which allow a manufacturer's products to compete with or complement a competitor's products—[that] promote the progress of the 'useful Arts.'"[165] Licenses "typically benefit everyone: the patent owner receives a steady stream of reasonable royalties from the entire industry using the standard, and consumers reap the benefit of a competitive playing field that would otherwise be severely constrained."[166] But holders of patents on interoperability standards can "hold an industry hostage by demanding crippling royalties."[167] Infringers

---

[156] Peter Lee, *The Evolution of Intellectual Infrastructure*, 83 Wash. L. Rev. 49 (2007).
[157] 547 U.S. 388 (2006).
[158] See MercExchange, L.L.C. v. eBay, Inc., 401 F.3d 1323, 1339 (Fed. Cir. 2005).
[159] *eBay*, 547 U.S. at 391.
[160] Id. at 395. Justices Breyer, Souter and Stevens joined this opinion.
[161] Id. at 396.
[162] Id.
[163] Id. at 396-97. The President of the EU has reportedly endorsed the discretionary issuance of injunctions in patent infringement cases based on considerations of equity. See IBM Corp., The Soft IP Agenda—A Viable European Community Patent, March 2008, at p. 1.
[164] See Brief of Amicus Curiae Nokia Corp. in Support of Petitioners, eBay, Inc. v. MercExchange, L.L.C.
[165] Id. at 4.
[166] Id. at 12.
[167] Id.

of patents on interoperability standards should be eligible for compensation under *eBay*, Nokia argued, but not injunctive relief.

Professor Lee did not rely upon the Nokia brief in support of his argument, but reaches much the same conclusion. Lee recommends that courts deny injunctions when "1) the infringed patent claims an infrastructural invention; 2) the infringer is actually using the patented invention in an infrastructural manner; and 3) the patented invention is not reasonably available through licensing."[168] Patents on interfaces essential for interoperability are among the infrastructural inventions that he thinks should meet this test.[169] Lee perceives his proposal to be "an action-forcing mechanism that will motivate patentees to come to the negotiating table and rationalize the balance of power once they get there"[170] because they will no longer have the leverage of an assured permanent injunction to obtain excessive rents for use of their infrastructural inventions. The public interest will be served, he argues, because the invention can be used to enable interoperability, but the patent holder will also be compensated for the use.[171]

Japanese policymakers have taken an alternative approach to creating a liability rule approach to patents essential to interoperability. The Japanese Ministry of Economy, Trade, and Industry (METI) established a "Study Group on the Legal Protection of Software and Promotion of Innovation," in 2005, which expressed serious concerns about the exclusionary potency of interface patents.[172] The Study Group's Interim Report noted that "[i]n the software sector, which is multi-layered, communication-enabled and with a tendency to have lock-in effects on users, the granting of patents may created unduly powerful exclusive rights."[173] Even though the Study Group recognized that most patents are exercised in a manner that promotes innovation, it asserted that interface patents posed risks of adverse effects on innovation.[174] The Study Group encouraged the use of Creative Commons-type licensing for patents affecting interoperability, but also recommended that compulsory licensing and enhanced application of anti-monopoly law should be considered as responses to such patents.[175]

Two years later, METI published its "Interpretive Guidelines on Electronic Commerce and Information Property Trading," which announced that a refusal to license patents essential for interoperability may constitute an abuse of intellectual property

---

[168] Lee, supra note xx, at 46.
[169] Id.
[170] Id. at 109. Another way that a liability rule could be implemented as to interface patents would be for the U.S. government to exercise its power to practice patented inventions and to authorize others to do the same subject to an obligation to compensate the rights holder for the use under 28 U.S.C. sec. 1498.
[171] See infra notes xx and accompanying text for a discussion of the viability of this option.
[172] Ministry of Economy, Trade, & Industry, Press Release, Interim Report of the Study Group on the Legal Protection of Computer Programs and Promotion of Innovation, Oct. 11, 2005, available at http://www.meti.go.jp/english/information/data/051011SoftInnove.html ("Interim Report").
[173] Id.
[174] Id.
[175] Id.

rights.[176] "Where a software provider holding a high market share has exclusive rights in connection with the technology related to interoperability/interfaces (even more significantly if such technology has been standardized), this tends to maintain the monopolized market conditions and undermines the incentives for innovation due to the adverse competitive effect."[177] The Guidelines make clear that whether a particular refusal to license an interface patent is an abuse of IP rights will be determined through a comprehensive assessment of the facts on a case by case basis, taking many factors into account.[178]

METI illustrated the potential for societal harm from interface patents with examples: patents that implicate interoperability of software that supports critical infrastructure, universal software that is widely used in society, and information services in which particular individuals participate, such as online auctions, where if the system is disabled by an interface patent, it will damage not only the developer of the information system, but also the operators of the online business and users of its services.[179]

A third liability rule approach to interface patents was proposed during the debate over the proposed European Directive on the patentability of computer-implemented innovations.[180] The Foundation for Free Information Infrastructure (FFII) urged the European Parliament to adopt its proposal to require owners of patents on interfaces indispensable to achieving interoperability to license such patents on reasonable and non-discriminatory (RAND) terms.[181]

A fourth liability-rule initiative that would affect, although it is not directly aimed at, interface patents is a proposed "Soft IP" right.[182] Under it, firms could apply for a European Community-wide patent without having to pay for the patent to be translated into all EU languages, but the Soft IP patent would only give the owner the right to compensation for use of the patent, not a right to exclude. Lawyers for IBM Corp. hope that this regime will be adopted and widely used by firms seeking patents on interfaces and other software-related inventions.[183]

D. Invoking Competition and Antitrust Law to Enable Interoperability

---

[176] Ministry of Economy, Trade, & Industry, Interpretative Guidelines for Electronic Commerce (March 2007), 192-193, 201, available at http://www.meti.go.jp/english/information/data/IT-policy/interpretative_guidelines_on_ec070628.pdf

[177] Id. at 193, n. 36.

[178] Id. at 196.

[179] Id. at 201, n. 51.

[180] See Bray, supra note xx.

[181] FFII Plenary Amendments at 2-3. This proposal was not included in the Council's May 2005 Common Position. Bray, supra note xx, at par. 27-28.

[182] See, e.g., James Nurton, *IBM Flies Soft-IP Community Patent Plan*, Managing Intellectual Property, July 31, 2007, available at http://www.managingip.com/Article/1398998/IBM-flies-soft-IP-Community-patent-plan.htm.

[183] Conversation with David Kappos, Vice President and Assistant General Counsel for Intellectual Property, IBM Corp., April xx, 2008.

Competition and antitrust authorities have sometimes scrutinized the practices of dominant firms that have thwarted, or attempted to thwart, the development of compatible technologies. The European Commission has twice invoked competition law as a regulatory tool for facilitating interoperability of ICT systems. The first arose when the Commission initiated a case against IBM Corp. in the 1980's, and the second when the Commission took action against Microsoft in the early years of the 21st century. The latter case resulted in the Commission ordering Microsoft to provide information necessary to interoperability to developers of competing systems. Microsoft was also forced to disclose interface information and license IP rights in interfaces as a remedy violating U.S. antitrust laws in the 1990's.

Although this Article will discuss the Microsoft cases at some length, a brief review of the IBM case is worthwhile, as it informed the Commission's intervention in favor of interoperability during the time that the Software Directive was being developed.[184] The IBM case also affected the Commission's perception of the competitive harms likely to flow from Microsoft's withholding of interface information and of the likely competitive benefits of forcing Microsoft to disclose interface information to its competitors.[185]

During the heyday of its dominance of the computer industry, IBM bundled its proprietary hardware, software, and peripherals together and treated interfaces as trade secrets. IBM's insistence that its customers buy bundled systems and its refusal to provide interface information to other firms impeded the development of interoperable components and systems. Even after IBM started unbundling software and peripherals, under pressure from antitrust authorities,[186] it did not publish its interfaces, but rather licensed them as trade secrets on royalty-bearing terms. Although licensing interface information to other firms did facilitate the development of IBM-compatible technologies, IBM upset its licensees by making frequent changes to its interfaces, which caused the licensees' previously compatible technologies to be less compatible or completely incompatible.[187] European competition law authorities charged IBM with abusing its dominant position by, among other things, changing interfaces in a manner that rendered IBM-compatible peripherals inoperable. IBM settled the lawsuit by

[184] The key sui generis provisions of the Software Directive—the recital's characterization of interfaces essential for interoperability as ideas and principles and Art. 6's authorization of decompilation for purposes of achieving interoperability—were legal innovations derived from the Competition Directorate's intervention in the policymaking process. The initial draft of the directive did not contain these provisions.
[185] See, e.g., F.M. Scherer, *Thinking About the European* Microsoft *Case*, 84 Antitrust & Trade Reg. Rep. 65, 65-66 (Jan. 23, 2003) (discussing similarities between the Commission's case against Microsoft and its earlier case against IBM over delayed disclosures of and changes to interfaces).
[186] Band & Katoh, supra note 32, at 22.
[187] See, e.g., Maria Lilla Montagnani, *Predatory and Exclusionary Innovation: Which Legal Standards for Software Integration in the Context of the Competition v. Intellectual Property Rights Clash?*, 37 I.I.C. 304 (2006) (discussing allegations that IBM had engaged in predatory innovation). See also In re IBM Peripheral EDP Devices Antitrust Litigation, 481 F. Supp. 965 (N.D. Cal. 1979)(peripheral makers charged that IBM's changes to interfaces were anticompetitive because they thwarted compatible products). Similar claims have been made against Microsoft. See CFI Decision, supra note 4, at parag. 282.

agreeing to pre-disclose changes to its interfaces to aid other firms in adapting their products in a timely manner.[188]

A decade or so after the IBM case, the European Court of Justice handed down the *Magill* decision, and later *IMS Health*, that established that a dominant firm's refusal to license IP rights can, in exceptional circumstances, constitute an abuse of dominant position under competition law.[189] *Magill* and *IMS Health* establish a four-part test for determining whether such exceptional circumstances exist: 1) the IP at issue must be indispensable for carrying on a particular business, 2) the refusal to license must be likely to eliminate competition in a secondary market, 3) the refusal to license must prevent the emergence of a new product for which there is potential consumer demand, and 4) the refusal was not objectively justified.[190]

The Commission applied and adapted this test in its competition law proceeding against Microsoft Corp. in the early 2000's.[191] Microsoft was charged with abuse of dominant position because it was unwilling to supply enough interface information and supporting technologies to enable Sun Microsystems to adapt its Solaris WGS-OS so that it could be fully compatible with—indeed, functionally equivalent to—Microsoft's Windows-based OS technologies, especially with the Active Directory technologies which coordinate communications amongst different units in distributed networked environments.[192] The Commission asserted that Microsoft had previously supplied a relatively high level of interface information to makers of WGS-OS technologies and that Microsoft had later withheld this information in order to gain additional market share at the expense of its rivals, thereby abusing its dominant position.[193]

---

[188] IBM also tried to stop firms, notably Fujitsu, from developing platforms capable of interoperating with applications written for IBM computers. See Band & Katoh, supra note 30, at 27-28. IBM twice charged Fujitsu with unlawful copying of IBM programs, including interfaces, although the cases settled. See, e.g., Stork, supra note 30. IBM's ability to prosper notwithstanding its facilitation of Fujitsu's competing platform seems to have given the Commission confidence that requiring Microsoft to supply interface information to its competitors would not undermine its ability to recoup its R&D investments.

[189] RTE & ITP v. Commission, 1995 ECR I-743 (ECJ); IMS Health v.NDC Health, 2004 ECR I-5039 (ECJ).

[190] See, e.g., Francois Leveque, *Innovation, Leveraging, and Essential Facilities: Interoperability Licensing in the EU Microsoft Case*, in ANTITRUST, PATENTS AND COPYRIGHTS: EU AND US PERSPECTIVES at 104 (F. Leveque & H. Shelanski, eds. 2005).

[191] Id. at 104-06.

[192] CFI Decision, supra note 4, at par. 2-3. Although the Commission's challenge relied heavily on *Magill* and *IMS Health*, it is worth noting that the Commission's charges were not directly based on Microsoft's unwillingness to license its IPRs in the interfaces, but rather about its unwillingness to supply detailed interface information to Sun and others.

[193] EC Decision, supra note 4, at par. 590-97. Sun and other makers of UNIX-related computer systems had been able for some years to get access to Windows OS interface information because Microsoft had licensed AT&T to have source code of previous iterations of its OS so that Unix systems could be compatible with Windows environments. When this arrangement started, Sun and Novell were among the leading makers of OS for what the Commission called work group server technologies. This was not a market Microsoft was initially in, which is why it made sense for Microsoft to supply interface information to AT&T for use in UNIX technologies. After Microsoft entered the market, it no longer had the same incentive to supply detailed information to AT&T, and the license deal ended. Updated versions of Microsoft's WGS-OS included Active Directory technologies that tightly coupled linkages between Windows and the WGS-OS. These were the technologies that Sun and others were particularly keen on

In March of 2004, the Commission found that Microsoft had a dominant position in the PC-OS market and that the information that Sun had sought from Microsoft was indispensable to its ability to remain a viable competitor in the WGS-OS market.[194] Microsoft's refusal to supply this information threatened to eliminate competition in the WGS-OS market because of powerful network effects that was tipping this market to Microsoft's product.[195] There was evidence that customers preferred many features of other WGS-OS systems, including Sun's; yet, customers also valued compatibility with Windows technologies so that Microsoft had this advantage over Sun.[196] Although there was no separate product whose emergence was being thwarted by Microsoft's product, as in *Magill*, the Commission adapted the *Magill/IMS Health* new product test by concluding that Microsoft's refusal to supply interoperability information was undermining Sun's ability to develop new features for its WGS-OS.[197] The Commission concluded that Microsoft had not shown that incentives to invest in innovation in the WGS-OS market as a whole would be undermined if Microsoft supplied the requested interface information.[198]

To remedy this abuse, the Commission ordered Microsoft to prepare sufficiently detailed interface specifications to enable Sun and other makers of WGS-OS systems to achieve interoperability with Microsoft's Windows-based technologies, to provide the specifications to Sun and others on reasonable and non-discriminatory (RAND) terms, and to update the information promptly as its interfaces changed.[199] Microsoft was also ordered to establish an evaluation mechanism to ensure compliance with the Commission's order.[200]

Microsoft appealed the Commission's order to the European Court of First Instance (CFI), arguing, among other things, that the Commission had misinterpreted the interoperability provisions of the Software Directive,[201] that its ownership of IPRs in the interfaces provided an objective justification for its refusal to supply extensive interoperability information to its competitors,[202] and that unless the company had

---

getting access to. While it is true that Microsoft had supplied interface information to Sun through AT&T in the past, it had not previously supplied information about the Active Directory technologies.

[194] EC Decision, supra note 4. The Commission also ruled against Microsoft on a separate charge as to abuse of dominant position in respect of media player software. Id. The Commission's ruling on media player issues is not relevant to this article.

[195] Prof. Leveque reasons that once Microsoft had attained a certain market share in the work group server OS market, it would have an interest in diminishing the supply of interface information; less interface information would cause its competitors' products to interoperate less successfully; this, in turn, would cause customers and ISVs to be concerned about being stranded. The market would then tip to Microsoft, with network effects to finish the work of killing off the competition. Leveque, supra note xx, at 113-14.

[196] CFI Decision, supra note 4, at xx

[197] Id. at xx

[198] Id. at xx

[199] Id. at 48.

[200] Id.

[201] Id. at par. 121.

[202] Id. at par. 124. Microsoft claimed copyrights in original interface documents, trade secret protection for the interfaces themselves, and patents on some communications protocols. Id.

freedom to choose how to exercise its IP rights in interfaces, it would have inadequate incentives to invest in research and development to improve its products.[203]

The sharpest difference between Microsoft and the Commission lay in their contrasting interpretations of the interoperability provisions of the 1991 Software Directive. The CFI characterized the difference as whether the Directive was intended to permit one-way or two-way interoperability.[204] Microsoft interpreted the directive as aimed at facilitating one-way interoperability, that is, as intended to facilitate interoperability between the program whose interface information was being sought (e.g., the Windows PC-OS) and complementary products (e.g., applications designed to run on Windows). The Commission's two-way theory posited that the Directive was also intended to facilitate development of functionally equivalent programs to the platform in question, such that the platform's competitors could successfully run programs that ran on the platform whose interface information was at issue.[205]

Microsoft argued that its existing licensing programs already enabled development of complementary products, which is all, in its view, that the Directive was intended to achieve.[206] It objected to being required to give competitors so much information as to allow them to "clone" its technologies.[207] Microsoft argued that this forced disclosure would be harmful to investments in innovation in the WGS-OS market. Microsoft itself would have little incentive to invest in innovation if it was forced to give its interfaces away to its competitors and be unable to benefit from the exclusive rights conferred by IP laws.[208] Microsoft also contended Sun and other competitors in the WGS-OS market would invest less in innovation because the Commission's order meant that they could benefit from the fruits of Microsoft's R&D without doing their own.[209]

In support of its two-way compatibility theory, the Commission pointed out that the critical distinction in the Software Directive is that between interfaces and implementations.[210] The Directive extends protection to the latter, but not to the former. Because its order did not require Microsoft to disclose source code, algorithms, or other

---

[203] Id. at par. 267-74.
[204] Id. at 108, 225-26.
[205] Hart characterizes one-way interoperability as enabling multi-vendor compatibility and two-way as enabling plug-replaceability. See Robert J. Hart, *Interoperability Information and the* Microsoft *Decision*, 2006 Eur. Intell. Prop. Rev. 361, 361. Hart asserts that the Directive was only intended to support multi-vendor compatibility. Id. Although Hart was actively engaged in developing the 1991 Directive, the text of the Directive does not support this theory. (Two other ways to characterize the one-way vs. two-way debate is to distinguish between vertical (aka one-way) and horizontal compatibility (aka two-way), see Weiser, supra note xx, at xx, or between complementary (aka one-way) and functional equivalent (aka two-way) products. See infra notes xx and accompanying text.)
[206] CFI Decision, supra note 4, at par. 121.
[207] Id. at par. 110. See, e.g., Page & Childers, supra note xx, at xx (explaining why Microsoft believed that disclosure of certain algorithms and other program internals would be necessary to achieve interoperability with the Active Directory technologies).
[208] Id. at par. 668.
[209] Id. at par. 670.
[210] Id. at par. 195-99.

internal design details of Microsoft's technologies, but only program interfaces,[211] it was not allowing competitors to clone Microsoft's technologies, but only to interoperate with them. In its view, the order merely required Microsoft to comply with the legislatively endorsed policy favoring interoperability embedded in the Software Directive.[212]

Although the Commission denied that Microsoft had any IP rights in its interfaces, [213] it also relied upon *Magill* and *IMS Health* as precedents holding that ownership of IP rights was not, of itself, an objective justification for refusal to license such rights.

Responding to Microsoft's investment disincentives argument, the Commission asserted that Microsoft would be able to recoup some of its R&D expenses from license fees the Commission had authorized it to charge Sun and others for disclosure of interoperability information, the price for which was depended in part on the level of innovation in the interfaces.[214] Microsoft and its competitors would have ample incentives to invest in further refinements and improvements in their technologies, apart from interfaces, in order to respond to and fuel consumer demand.[215] Because Sun and others had invested in innovative WGS-OS designs prior to Microsoft's decision to cut back on its previously higher level of interoperability disclosures, the Commission believed that requiring Microsoft to disclose the requested interoperability information would not significantly dampen its or its competitors investments in innovation in the future.[216]

In September 2007, the CFI affirmed the Commission's order, holding, (1) that the Commission's interpretation of the Software Directive was sound,[217] (2) that under *Magill* and *IMS Health*, ownership of IPRs was not, by itself, an objective justification for refusal to license them to others, (3) that the exceptional circumstances required by *Magill* and *IMS Health* had been met,[218] and (4) that Microsoft had failed to prove that its

---

[211] Id. at par. 148, 204.

[212] Id. at par. The Commission's first experience with the competitive effects of refusals to disclose interface information was in its action against IBM, the dominant firm in the computer industry in the 1960's-1980's, for refusing to disclose changes in interfaces to manufacturers of plug-compatible products (such as disk drives and tape storage devices) and makers of IBM-compatible systems. This Commission's proceeding was eventually settled by IBM's agreement to disclose interface information to other firms in advance of releasing new systems into the market. See Scherer, supra note xx, at 65-66. This experience informed the Commission's perspective on the interoperability provisions of the Software Directive. See EC Decision, supra note 4, at parag. 737-40 (discussing the Commission's competition law challenge of IBM for abuses in disclosure of interface information).

[213] The Commission asserted that any copyright that Microsoft might claim in interface specification documents would not be infringed by other firms' implementing the interfaces in independently written programs. The Commission seems to have questioned whether Microsoft's protocols were innovative enough to qualify for patent protection. Microsoft's trade secrecy claims were discounted as not being intellectual property rights; the value in them lay in their secrecy, not in any innovation they might embody. Id. at 276-80.

[214] CFI Decision, supra note xx, at par.

[215] Id. at par.

[216] Id. at par. .

[217] Id. at par.

[218] Id. at xx.

incentives to invest in innovation would be diminished by the order, observing that the firm had provided only vague, general, and theoretical arguments in support of this claim.[219] The CFI pointed out that it was standard industry practice to license interface information,[220] that Microsoft itself had agreed to provide interface information in settling litigations against it in the U.S. The order in this case was, moreover, consistent with the Software Directive and the IBM settlement in a similar competition case in the mid-1980's.[221] Microsoft decided not to appeal the CFI ruling.[222]

Although Microsoft disclosed considerable amounts of interface information under the order,[223] the Commission has fined Microsoft $1.35 billion for failing to provide sufficiently detailed interface information.[224] Microsoft and the Commission have had on-going disagreements over the level of innovation embodied in its interfaces—Microsoft, unsurprisingly, claims to have developed very innovative interfaces, and the Commission has argued they are mundane—which affects the price which Microsoft can charge licensees for providing interoperability information.[225]

Key differences between U.S. and EU antitrust/competition law cast doubt on whether U.S. antitrust would pursue or U.S. courts would uphold similar claims against Microsoft. One of the two theories underlying the Commission's proceeding against Microsoft seems to be that Microsoft's interfaces are an "essential facility" which Microsoft, as the dominant firm with control over access to that facility, was obliged, by virtue of its market power, to allow others access on fair and non-discriminatory terms as long as doing so would not cause undue congestion or the like in providing access to that facility.[226]

The viability of the "essential facility" doctrine of U.S. antitrust law is unclear after the U.S. Supreme Court decision in Verizon Communications v. Law Offices of Curtis V. Trinko.[227] Even assuming that *Trinko* did not deliver a death blow to that doctrine,[228] U.S. courts would likely be more sympathetic to Microsoft's claims that it was justified in refusing to license interface information because of its IP rights in the

---

[219] Id. at par. 689-90.
[220] Id. at xx. Neither the Commission nor the CFI considered whether licensing interface information to developers of complementary practices was more common than licensing this information to makers of functionally equivalent products. I suspect the former is far more common than the latter.
[221] CFI Decision, supra note 4, at par. 702-10.
[222] See Microsoft Corp., Freedom to Innovate Newsletter, *European Officials Question Value of Microsoft's Innovation*, available at http://www.microsoft.com/freedomtoinnovate/newsletter/finnews_033007.aspx.
[223] See Stephen Castle, *Microsoft Gets Record Fine and a Rebuke From Europe*, New York Times, p. C3, Feb. 28, 2008 (reporting that Microsoft had published 30,000 pages of previously secret source code for the Windows operating system to comply with the Commission's order; yet the Commission considered this disclosure inadequate).
[224] Id.
[225] CFI Decision, supra note xx, at par.
[226] Leveque, supra note xx, at 120-21 (discussing the Commission's essential facility theory).
[227] 540 U.S. 398 (2004).
[228] See, e.g., Michael A. Carrier, *Refusals to License Intellectual Property After* Trinko, 55 DePaul L. Rev. 1191 (2006).

interfaces and that unless it is able to recoup its R&D expenses, it will have too little incentive to invest in innovation. U.S. courts have thus far been unwilling to hold that a refusal to license IPRs to competitors is a violation of the U.S. antitrust laws.[229]

There is, however, some similarity between the EU *Microsoft* case and the Supreme Court's ruling in Aspen Skiing Co. v. Aspen Highlands Skiing Corp.[230] In both cases, there was a history of sharing of resources by competitors that had grown the market for both; at some point the dominant party withdrew from cooperation in a manner that seemed to lack an independent business justification.[231] It is conceivable that U.S. courts, in an appropriate case, would take into account that network effects in the software industry can tip the market to a single provider, not so much because of the intrinsic innovation in its interfaces, but rather because network effects kick in once the developer's interface becomes a de facto interoperability standard. *Trinko* relies heavily on the notion that it is important to preserve incentives to invest in creating the market for the facility said to be essential, and that is as it should be. In network industries, however, network effects themselves may provide powerful incentives for firms to invest in becoming the de facto interface standard.

While a refusal to license IP rights has never, of itself, been deemed an antitrust violation in the U.S., courts have sometimes ordered antitrust violators to license IPRs and/or disclose non-public information, such as interface specifications, to competitors.[232] The consent decree settling the U.S. antitrust case against Microsoft in the 1990's, for example, required Microsoft to disclose interface information and license IPRs in them to firms that might want to use them, even though U.S. antitrust authorities had not charged Microsoft with having misused any patents on its interfaces or refused to license IPRs in its interfaces to competitors.[233] U.S. antitrust officials persuaded the judge that oversaw the settlement to take a forward-looking approach to guarding against possible efforts by Microsoft to maintain the firm's monopoly in the Windows OS market by restricting access to interface information.[234]

One way to give other firms more access to Windows' interfaces would be to require Microsoft to license its source code to those who wanted to make interoperable programs.[235] This would certainly have lowered barriers to interoperability, but some expressed concern this remedy was more generous to competitors than was warranted by

---

[229] See, e.g., Herbert Hovencamp, Mark D. Janis, & Mark A. Lemley, *Unilateral Refusals to License in the U.S.*, in ANTITRUST, PATENTS AND COPYRIGHTS: EU AND US PERSPECTIVES at 104 (F. Leveque & H. Shelanski, eds. 2005).
[230] 472 U.S. 585 (1985).
[231] Id. at 609-11.
[232] See, e.g., William E. Kovacic, *Designing Antitrust Remedies for Dominant Firm Misconduct*, 31 Conn. L. Rev. 1285, 1304 (1999) (giving examples of licenses induced by antitrust oversight).
[233] See Page & Childers, supra note xx, at 83 (discussing this "forward-looking" approach). Some states that also sued Microsoft for antitrust violations had sought more comprehensive disclosure of interface information, id. at 90-91, but the judge overseeing these cases rejected the more expansive interface disclosure request. Id. at 103-08.
[234] Id. at 83.
[235] Jonathan Zittrain, *The Un-Microsoft Un-Remedy: Law Can Prevent the Problem That It Can't Solve Later*, 31 Conn. L. Rev. 1361, 1371-72 (1999).

the antitrust violations that had been found.[236] A second was to require licensing and disclosure of interface information to enable other firms to develop interoperable systems. Microsoft agreed to the latter in settling the U.S. antitrust case.[237]

Compulsory licensing of IPRs and/or knowhow, such as interface information, is challenging as an antitrust remedy because it requires close oversight as to exactly which IPRs must be licensed, how much detailed information must be transferred, how timely upgrade information must be provided, and how long the duty to license IPRs or supply information will need to last,[238] the very problems with which the Commission has been grappling in its recent proceedings against Microsoft.[239] Despite costly efforts to facilitate greater interoperability through forced disclosure of interface information, it does not appear that the U.S. order has accomplished the objective that the disclosure requirement was intended to achieve: more competition in the PC-based OS market.

At the time when remedies in the U.S. v. Microsoft case were being hotly debated, some commentators were skeptical that requiring Microsoft to disclose interface information would reduce its dominance in the OS market. Some argued instead for a structural remedy, such as by breaking Microsoft up into one that developed OS software and another that developed applications; the former could then license the OS interface specifications to the latter on equal footing with other applications providers.[240] Among the difficulties with this proposal was that it would require making difficult judgments about what "belongs" in an OS and what "belongs" in applications, which judges and antitrust officials lacked expertise to assess.[241]

Alternatively, some favored ordering Microsoft to break up into multiple companies ("baby Bills"), each of which would develop Windows OS technology and license interfaces to applications providers. This might produce more competition in the OS market, but it risked fragmentation of the OS market, which would lead to greater costs and a loss of benefits to consumers and applications developers of a single de facto interface standard.[242]

---

[236] Id.
[237] U.S. v. Microsoft Corp., 224 F. Supp.2d 76, 269 (D.D.C. 2002).
[238] Kovacic, supra note 259, at 1304.
[239] Connecticut Law Review published a symposium issue of articles on the remedy challenges posed by the U.S. v. Microsoft case in 1999. See Roger M. Langer, *Symposium Introduction*: U.S. v. Microsoft Corp., 31 Conn. L. Rev. 1245 (1999). See generally Software & Info. Indus. Ass'n, *Addressing the Microsoft Challenge--Restoring Competition to the Software Industry* (Feb. 1999), available at http://www.manishin.com/pressdocs/siia.pdf (providing a comprehensive analysis of remedy alternatives).
[240] See, e.g., Zittrain, supra note xx, at (discussing the source code disclosure proposal).
[241] Id. at 1370-71.
[242] See, e.g., Stan J. Liebowitz, *Breaking Windows: Estimating Some Costs of Breaking Up Microsoft Windows* (2000), available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=214312. But see Robert J. Levinson, R. Craig Comaine, Steven Salop, *The Flawed Fragmentation Critique of Structural Remedies in the Microsoft Case* (2000), available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=204874. See also Stan J. Liebowitz, *A Fool's Paradise: The Windows World After a Forced Breakup of Microsoft* (2000), available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=218178 (responding to Salop et al.)

Yet another alternative for resolving conflicts over IPRs in and disclosures of interface information was invented in the mid-1980's during an arbitration of a IPR dispute between IBM Corp. and Fujitsu over the OS software that Fujitsu made that was fully compatible with (and a functional equivalent to) IBM's OS for its System 360/370 computers.[243] Fujitsu had sold IBM-compatible OS software for mainframe computers without objection from IBM from the mid-1970's until 1982. Then IBM charged Fujitsu with having misappropriated intellectual property rights in its OS. Fujitsu asserted that it had only appropriated public domain and unprotectable elements from IBM's programs. Although IBM and Fujitsu settled this first dispute in 1983, key terms were left undefined and the compromise soon broke down. An arbitration ensued. One of the many difficulties the arbitrators faced was that the availability and scope of copyright protection in computer programs was unclear at that time. Rather than attempting to resolve this IPR issue, the arbitrators proposed a forward-looking solution, a key element of which was a "clean room" approach to obtaining essential interface information.

Under the regime established during the IBM-Fujitsu arbitration, IBM, in exchange for an agreed upon royalty payment from Fujitsu, was obliged to deliver source code for any new releases of its OS to a secure facility operated by a special set of Fujitsu employees.[244] Fujitsu's "clean-room" team would then analyze the source code and extract interface information. Upon compiling the information essential to Fujitsu's ability to continue to develop IBM-compatible OS software, IBM sent a team to review the compiled interoperability information. When it signed off that Fujitsu's team had only extracted interface information, not other innovations in the IBM software, the clean-room team would then transfer the interface information to the Fujitsu OS development team so that they could reimplement the interfaces in Fujitsu's own independently developed programs.[245]

In a suitable antitrust case involving misuses of IPRs in interfaces and refusals to disclose interface information, a court might want to consider a similar "clean room" and licensing regime as that which settled the IBM-Fujitsu dispute more than two decades ago. This is admittedly a costly way to facilitate interoperability, but it does have some advantages: it avoids wasteful rounds of bickering over how much information the firm must disclose and it places some of the burden of obtaining the information on the firms that want to develop functional equivalent programs.[246]

E. Private Regulation of Interface Patents.

In large part because many industry participants are aware of the high exclusionary power of interface patents, several private sector initiatives have focused on

[243] The facts recited in this paragraph were taken from Stork, supra note xx.
[244] Id. See also Robert H. Mnookin, *Creating Value Through Process Design: The IBM-Fujitsu Arbitration*, 47 Arbit. J. 6 (Sept. 1992).
[245] Id. at 11. The arbitrators retained authority to resolve any further dispute between IBM and Fujitsu over the exchange of source code and interface information; there were, however, no further disputes, as the parties had adequate incentives to cooperate with this procedure. Id.
[246] See Page & Childers, supra note xx, at 117-21 (discussing many difficulties encountered in determining how much information Microsoft was obliged to disclose under the consent decree).

development of policies that will ensure that patents on ICT interfaces will be exercised in a manner that will promote interoperability rather than thwarting it. This is especially important when an interface technique, such as a communications protocol, is under consideration for formal adoption as a standard.

The most significant of these private sector initiatives has been a policy promulgated by the World Wide Web Consortium (W3C) which requires member firms to agree that if they own patents that "read" on any standard adopted by W3C that is essential to interoperability on the Web, those patents must be licensed on a royalty-free (RF) basis.[247] The initial impetus for adoption of this policy was a patent holder's claim that the W3C's Platform for Privacy Preferences (P3P) standard infringed its patent. Although the W3C concluded that the P3P standard did not infringe that patent, senior officials realized that the W3C would likely be faced with other patent claims affecting its standards. After thorough deliberations, the W3C decided to adopt an RF policy as to standards essential to Web interoperability, concluding that this policy was the optimal way to promote the continued progress of the open Web.[248]

Unlike the W3C, the Organization for the Advancement of Structured Information Standards (OASIS) does not mandate RF licensing of interface patents held by member firms that are proposed as standards. OASIS was, however, influenced by the W3C policy, for it developed two RF licensing options for technical committees (TCs) operating under OASIS' aegis to adopt, although it also allows TCs to adopt policies that commit holders of patented technologies adopted as standards to licensing them on reasonable and non-discriminatory (RAND) terms.[249] OASIS now requires TCs to announce at the time of TC formation which IP policy they have adopted. Interestingly, the overwhelming majority of TCs formed since this new policy was put in place have adopted RF policies for applications and webservices standards approved by OASIS.[250] Patents on interface components of OASIS standards are, therefore, generally available on open terms.

Although RF policies for interface patents do not make such patents unenforceable, they substantially reduce the leverage that the patents would otherwise provide their owners as well as their economic value. This, in turn, dampens incentives to acquire such patents. Free and open source developers nonetheless sometimes object

[247] W3C Patent Policy, Feb. 4, 2005, available at http://www.w3.org/Consortium/Patent-Policy-20040205/. The policy does, however, contain a procedure whereby one can attain an exclusion from the RF commitment. See id. at part 4.

[248] See Testimony of Daniel Weitzner, Technology and Society Domain Leader of the World Wide Web Consortium, at Joint Hearings on Competition and Intellectual Property Law and Policy in the Knowledge-Based Economy: Standards and Intellectual Property: Licensing Terms Before the Department of Justice and Federal Trade Commission, April 18, 2002, available at http://www.ftc.gov/opp/intellect/020418weitzner.shtm.

[249] See OASIS Intellectual Property Rights Policy, http://www.oasis-open.org/who/intellectualproperty.php.

[250] Conversation with Robert J. Glushko, OASIS Board member, March 1, 2008.

to W3C and similar RF policies because the these licenses include some restrictions that are incompatible with the practices of this community.[251]

Like the W3C and OASIS, the overwhelming majority of SSOs that adopt standards affecting the ICT industry require members who participate in standard-setting processes to disclose patents that are essential to any standard under consideration by that SSO.[252] Most also require a pre-commitment to licensing such patents on RAND terms,[253] although what "reasonable" means is sometimes hotly debated. Thus, at least interface patents that have been adopted as standards will generally be available under RAND licenses, even where they are unavailable under RF terms.[254]

Another private initiative that fosters interoperability in the patent-intensive landscape of the ICT industry is the Open Invention Network (OIN), a patent pool recently formed by several major IT industry firms to build a portfolio of software patents that support open source software projects. OIN "acquires patents and makes them available royalty-free to any company, institution or individual that agrees not to assert its patents against the Linux System."[255] The OIN pool acquires software patents of all kinds, including some that cover valuable interfaces.[256] Other similar pools seem to be forming.[257] Some firms are, moreover, making unilateral commitments not to enforce certain interface patents.[258]

In addition to the patent pools and unilateral commitments mentioned above, it is a common practice in the ICT industry for firms to cross-license their patent portfolios.[259] Some interface patents will be components of these portfolios. The pervasiveness of cross-licenses in the ICT industry is yet another check on potentially abusive exercise of patents. These and other private initiatives cannot, of course, blunt the force of all interface patents that might impede interoperability, which is why some nations have adopted or considered more interventionist strategies.

III. What Is the Right Policy Response to Interface Patents Essential to Interoperability?

---

[251] See, e.g., FSF's Position on the W3 Consortium "Royalty Free" Patent Policy: Our Position, rewritten 1 June 2003, available at http://www.gnu.org/philosophy/w3c-patent.html (expressing objection to field of use restrictions and restrictions on implementation of the specification precisely as set forth in the license).
[252] Lemley, supra note xx, at 1904-05.
[253] Id. at 1906.
[254] The European Telecommunications Standards Institute has an IP policy that makes interface standards available on RAND terms. See http://www.etsi.org/website/AboutETSI/LegalAspects/IPR_Policy_FAQ.aspx
[255] See http://www.openinventionnetwork.com/patents.php.
[256] The Meltzer patent, discussed supra note xx, for example, is part of the OIN pool.
[257] See, e.g., Francois Leveque & Yann Meniere, *Copyrights vs. Patents: The Open Source Software Legal Battle*, 4 Rev. of Econ. Res. on Cop. Issues 27, 42-43 (2007); Presentation of Hank Barry, Berkeley Center for Law & Technology, Intellectual Property and Entrepreneurship Conference, March 8, 2008.
[258] Leveque & Meniere, supra note 177, at 43.
[259] See, e.g., Gideon Parchomovsky & R. Polk Wagner, *Patent Portfolios*, 154 U. Pa. L. Rev. 1 (2005).

Anyone who reviews the rather extensive literature on the policy options for responding to the exclusionary potency of ICT interface patents would get the impression that patents on interfaces are very serious impediments to interoperability. As Part II has revealed, this literature offers many suggestions for regulating patents in order to foster more compatibility among ICT systems. Within this framing, the only question seems to be which of the numerous options would best achieve this objective.

If one begins the inquiry instead by surveying the vast array of ICT systems deployed in the modern world, it becomes evident that interoperability is quite prevalent, even if not ubiquitous.[260] Market incentives, largely driven by the potential power of network effects, are strong enough to induce many firms to publish ICT interfaces and make them available without IP restrictions or to make interface information available on licensing terms that are widely viewed as unobjectionable. Intra-industry cross-licensing of patents is, moreover, very common within the ICT industries, and many interface patents get licensed thereby. Insofar as patented interfaces have been adopted as standards by SSOs, they are likely to be available under either RF or RAND terms. The more fundamental interfaces are to the functioning of key infrastructures, such as the World Wide Web and webservices, the more likely the patents are to be available on RF terms.[261]

This is not to say that patents on interface designs never impede interoperability. Nintendo, after all, was able to bar Atari Games from making and selling games for its NES console because of its patent on an authentication mechanism.[262] That case was, however, decided fifteen years ago. Very few reported cases since then have involved ICT interface patents. There is some anecdotal evidence exists of open source software projects that were blocked by interface patents.[263] Yet, some open source projects have been able to implement patented interfaces. Samba and Microsoft, for example, were able to reach an agreement on Samba's use of certain network communications protocols.[264] (It is fair to observe that Microsoft's willingness to license these protocols to Samba on GPL-friendly terms was in large part due to the Commission's oversight of

---

[260] See, e.g., Gasser & Palfrey, supra note 1, at 6.

[261] It is, of course, possible that owners of patents on interface techniques may not be members of the W3C, OASIS, or other SSOs, and hence not committed to the RF or RAND policies of those organizations. Thus, private initiatives to foster interoperability through SSO policies may not be a complete solution to the interoperability problem. When SSOs learn of a patent that might "read" on an interface design that is under consideration as a standard, they may be able to design around it. There is, of course, greater risk of holdup if the design has already become a standard, and industry leaders have already implemented the design in their products. In circumstances where irreversible commitments have been made to an interface standard covered by an outsider's patent, courts may award reasonable royalties for infringement instead of injunctive relief. But see infra note xx and accompanying text for a discussion of court rulings on this.

[262] See supra notes xx and accompanying text.

[263] See, e.g., Andy Tai, *Microsoft prohibits GPLed work via licensing of CIFS standards*, Advogato, posted Apr. 5, 2002 at 07:27 UTC, available at http://www.advogato.org/article/453.html.

[264] Microsoft had initially been willing to license patents on two network communications protocols (U.S. Patent No. 5,264,261 and 5,437,013) to Samba on RF terms. Samba insisted it must be able to use the General Public License (GPL) for its software. See, e.g., Mary Jo Foley, *Microsoft and Samba Finally Come to Terms Over Windows Protocols*, ZDNet, Dec. 20, 2007, available at http://blogs.zdnet.com/microsoft/?p=1064. The final agreement allowed this.

its interface licensing practices.[265]) It is also fair to conclude that the W3C, OASIS, and METI would not have undertaken their policy initiatives concerning patents affecting interoperability if these organizations thought that such patents would never or only rarely presented impediments to interoperability.

Nor is it to say that patents could not become a bigger impediment to interoperability over time. FFII, for example, has expressed concern that Microsoft will undo the European Commission's ruling by seeking ever more patents on interfaces.[266] Yet, even if Microsoft does get more patents on interface techniques, there is reason to believe the Commission will be no more deferential to those patents than it was to the patents Microsoft raised in the 2004 proceeding. Microsoft's arguments that a refusal to license technological IPRs (i.e., patents on interfaces) should be treated more favorably than a refusal to license copyrights, as in *Magill* and *IMS Health*, failed to persuade the CFI. The Commission and the CFI are obviously comfortable with interpreting competition law as an appropriate means to patrol a dominant party's use of patents to impede interoperability.

I agree with Professor Lee's assertion that the Supreme Court's *eBay* decision might embolden courts to award reasonable royalties to owners of interface patents instead of issuing injunctions for interfaces essential to interoperability. I further agree that this approach is more likely to be employed if the patents implicate widely used or key infrastructures.[267] Yet, it is worth noting that the Federal Circuit was unreceptive to Vonage's argument that in the interests of those who used VOIP through Vonage, the court should award damages, but not injunctive relief,[268] so it is far from certain that the Federal Circuit court would follow Lee's prescription. The post-*eBay* caselaw adds to this doubt,[269] for courts seem to be withholding injunctive relief thus far only in cases that appear to involve "patent trolls."[270] The *eBay* decision, however, suggests that withholding injunctive relief should not be limited to patent-troll-like cases. Nokia's amicus brief in *eBay* makes a powerful argument about why injunctive relief should be withheld in patent cases involving interfaces essential to interoperability.

---

[265] See, e.g., William H. Page & Seldon J. Childers, *Bargaining in the Shadow of the European* Microsoft *Decision: The Microsoft-Samba Protocol License* (May 2008), available at http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1117641.

[266] See FFII, *Microsoft Will Trump EU Competion Ruling with Patents*, Sept. 17, 2007, available at http://press.ffii.org/Press_releases/Microsoft_will_trump_EU_competition_ruling_with_patents.

[267] See supra notes xx and accompanying text. Lemley suggests that courts can use liability rules post-*eBay* when a patentee is seeking excessive returns in exchange for licensing a patent for a standard; however, he assumes that in a majority of cases involving standards, injunctive relief will probably issue. Lemley, supra note xx, at 167.

[268] *Vonage*, 503 F.3d at 1310-11.

[269] See, e.g., Benjamin H. Diessel, Note, *Trolling for Trolls: The Pitfalls of the Emerging Market Competition Requirement for Permanent Injunctions in Patent Cases Post*-eBay, 106 Mich. L. Rev. 305, 312-15 (2007) (presenting tables of cases).

[270] Id. at 318-22. Courts may also be wary of damage awards in lieu of injunctions because of the difficulties that attend price-setting by non-market actors through compulsory licenses, especially as applied to developers of competing platforms. Id. at 342-43.

I do not foresee any U.S. court ruling that a refusal to license an interface patent constituted patent misuse, even if the claim was wrapped in the language of tying arrangements.[271] The U.S. patent statute specifically provides that refusing to license patents is not misuse.[272] Nor do I foresee that U.S. courts will treat a refusal to license an interface patent or to disclose interface information as an antitrust violation, even if the license or disclosure is necessary to achieving interoperability. *Trinko* calls into question the possibility that U.S. courts would treat ICT interfaces are essential facilities, refused access to which violates antitrust law.[273] Of course, an antitrust violator may be required to license patents and provide interface information as a remedy, as in U.S. v. Microsoft.[274]

This is not to say that U.S. patent misuse and antitrust policy would not adapt if patents on interfaces become a substantial impediment to interoperability in the future.[275] There is, however, too little evidence that patents are a major impediment to interoperability to justify serious consideration of stronger policy measures such as excluding interfaces from patentable subject matter or immunizing the use of patented interfaces to achieve compatibility.

While I have sometimes questioned whether patents are a suitable form of intellectual property protection for computer programs,[276] I believe that if patents *are* available for software inventions, it is difficult to justify withholding them from interface designs as they meet patent law's novelty, nonobviousness, and utility standards. Insofar as the European Commission believes that interface techniques can never satisfy patent law's invention standards,[277] I believe that the Commission is wrong. Interfaces enable new features, and they are thereby connected to the innovative features they enable.[278]

---

[271] See supra note xx and accompanying text for Cohen's argument as to why the Nintendo lock-out system was patent misuse. But see, e.g., In re Independent Service Organization Antitrust Litigation, 203 F.3d 1322 (Fed. Cir. 2000)(patentee can lawfully refuse to license third parties who want to make replacement parts). Treating lock-outs as misuse is perhaps less likely after enactment of the anti-circumvention rules of the Digital Millennium Copyright Act, which are codified at 17 U.S.C. sec. 1201, which privileges use of TPMs to protect copyrighted works. See Randal C. Picker, *Copyright and the DMCA: Market Locks and Technological Contracts*, in ANTITRUST, PATENTS AND COPYRIGHTS: EU AND US PERSPECTIVES at 104 (F. Leveque & H. Shelanski, eds. 2005). Yet, cases like Chamberlain Group, Inc. v. Skylink Techn., Inc., 381 F.3d 1178 (Fed. Cir. 2004), which rejected claims that bypassing a TPM for garage door openers violated sec. 1201, have interpreted the anti-circumvention rules narrowly to preclude lock-out TPMs that do not lead to copyright infringements. See also Dan L. Burk, *Anti-Circumvention Misuse*, 50 UCLA L. Rev. 1095 (2003).

[272] 35 U.S.C. sec. 271(d)(5).

[273] See supra notes xx and accompanying text.

[274] See supra notes xx and accompanying text.

[275] U.S. courts have not considered the special issues that arise from the power of network effects, which make proprietary control over interfaces so powerful. Insofar as patent and antitrust law become concerned with consumer harms arising from exclusionary uses of patents, this might justify U.S. policymakers moving toward an EU-style regulation of how dominant parties exercise IPRs.

[276] See Benson *Revisited*, supra note xx; Manifesto, supra note xx.

[277] See supra note xx and accompanying text.

[278] I credit Robert Barr for making this point. Barr also pointed out that if engineers at Internet Engineering Task Force meetings spend hours discussing the most "elegant" solution to a particular technical problem when setting standards for interfaces, there must be some innovation that distinguishes one solution from another.

Yet, I believe it would be rational for a nation to decide, as the EU arguably did, that interfaces essential to interoperability are "ideas" or "principles" that should be unpatentable as a sui generis matter. There is reason to question whether patent incentives are necessary to bring interfaces into existence. Developing software without interfaces is like building a house without windows or doors.[279] It is implausible that firms such as Microsoft would cease developing interfaces or improving interfaces if they could not patent them. The EU could reasonably have decided that it was giving such expansive protection to software internal designs, except for interfaces, and protecting developers against reverse engineering except when they were motivated by compatibility considerations. This expansive protection of a sui generis regime might have more than made up for giving no legal protection whatsoever to interfaces.[280]

Although patents on interface techniques may be patentable in the U.S., one can hope that it will become easier to weed out "bad" patents on arbitrary or trivial interface designs through the reinvigoration of the invention standard after the Supreme Court's *KSR* decision and other patent reforms (such as post-grant review).

Because firms have incentives to patent interface designs for reasons having less to do with innovation and more to do with excluding competitors from the marketplace, patent examiners should recognize that patents on interface techniques confer a high degree of exclusionary power. Patent examiners would be well advised to review claims for patents on interface designs and functionality more rigorously than they do as to other kinds of claims and to require more meaningful disclosure of interface designs if substantive standards of patentability are met.

There are, of course, some sectors where there is less interoperability than some might wish for (e.g., digital music services).[281] Patents on interfaces do not, however, seem to be the principal impediments to interoperability in those sectors. As Part I pointed out, firms sometimes choose non-interoperable strategies for their products and services, as Apple did with iTunes, in order to build their own network and reap the rewards of network effects that need not be shared with other competitors (e.g., RealNetworks). Trade secrecy and the ability to re-engineer interface components if a competitor successfully reverse engineers and implements an earlier interface in its products are more powerful protections for firms whose business models are built on non-interoperable strategies.

When it was considering the Software Directive, the EU could have required software developers to disclose interface information rather than just treating interfaces as unprotectable ideas or principles. The Commission may have believed it had done

---

[279] QuickStudy: Application Programming Interface (API), Jan. 10, 2000, available at http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=43487.

[280] Of course, if patent protection was unavailable for interfaces, developers might rely more on trade secrecy protection.

[281] France has undertaken regulatory efforts to bring about greater interoperability as to digital music. See, e.g., L.C. Angell, *French iTunes Interoperability Law Goes Into Effect*, Aug. 2006, available at http://www.ilounge.com/index.php/news/comments/french-itunes-interoperability-law-goes-into-effect/.

enough to induce disclosure of interface information by authorizing reverse engineering for interoperability purposes unless the information was licensed. The Software Directive does not, however, provide any remedy to would-be reverse engineers if interface information cannot be obtained in this manner. With very large and complex programs, such as Windows-based technologies involving distributed network resources and "cloud" computing, reverse engineering has become an infeasible way to get access to interface information. [282] Even when reverse engineering enables second comers to obtain interface information, its developer can change the interface relatively easily and thereby thwart an unlicensed competitor from attaining more than a short-run compatibility advantage.[283] This too is not regulated by the Software Directive.

It would, of course, be possible for national authorities to require developers of interfaces to be more forthcoming about their innovations, either as a sui generis matter or as a precondition of getting patent protection for interface techniques. Although firms prefer to maintain interface information as a secret, it is unclear why this information should be more sacrosanct from disclosure mandates than other business secrets.[284]

Although lack of access to interface information is a common impediment to interoperability, it is important to recognize that achieving interoperability is a more complex technical goal than legal commentators often acknowledge. Even when one has plentiful access to APIs, it can still be very difficult to achieve interoperability.[285] Interoperability typically requires a fine-grained agreement on syntax and semantics, and firms that want to interoperate must be precise on each dimension.[286] Proprietary interfaces are, moreover, often more difficult to implement than published or licensed APIs because the proprietor may, whether intentionally or inadvertently, have used or

---

[282] CFI Decision, supra note xx, at xx.

[283] See, e.g., Bill Rosenblatt, *RealNetworks and Motorola Open Apple iTunes/iPod Stack*, DRM Watch, available at http://www.drmwatch.com/drmtech/article.php/3387481. Thereafter, Apple changed its interface and undid RealNetworks temporary interoperability advantage.

[284] It will, for example, be interesting to see if France's effort to regulate disclosure of interface information for digital music systems succeeds in promoting greater interoperability in the digital music player sector.

[285] At least four kinds of technical problems can impede interoperability: differences in content, differences in encoding, differences in structure, and differences in semantics. Consider, for example, the following impediments to interoperability for electronic documents representing $100. If X represents $100 as <A>USD 100</A> and Y presents the same concept as <A>One Hundred US Dollars</A>, the discrepancy between them means that messages about $100 cannot interoperate. If X encodes $100 as <Amount>USD 100</Amount> and Y encodes it as USD,100, the differences in encoding will similarly prevent interoperability. If X encodes $100 as <Amount> USD 100</Amount> and Y represents it as </Currency> USD </Currency><Amount>100</Amount>, structural differences will thwart interoperability. Finally if X represents $100 as <Amount>USD 100</Amount> and Y represents this as <Price>USD 100</Price>, semantic differences will thwart interoperability. Glushko & McGrath, supra note 6, at 173.

[286] If, for example, an interface calls for location information, interoperability may not occur if the system is expecting GPS coordinates and the application has location information in postal codes. An API for obtaining weather information may not be precise enough if the entity from which it is seeking weather data has separate categories for temperature, humidity, windspeed, and the like. With interfaces, details like this matter a lot.

ambiguous terms in its specifications. [287] Yet, even when interface designs have been well-documented and standardized, firms may still need to develop guidelines and obtain expert advice about how to comply with them, and perhaps even have access to test suites and other tools in order to get the details right.[288]

There are nonetheless some policy measures—short of mandating interface disclosure—that would facilitate more access to interface information without undermining incentives to invest in innovative interfaces. Nations might, for example, emulate the EU Software Directive by treating license terms that forbid reverse engineering as a legal nullity, at least insofar as the reverse engineering is for purposes of seeking information essential to interoperability.[289] Nations might also recognize or create a privilege in patent law, akin already recognized in copyright, anti-circumvention, and trade secrecy laws, to permit reverse engineering for purposes of getting access to interface information.[290]

There is finally the question about whether refusals to license interface information or interface patents should be treated differently depending on whether the would-be interoperator plans to develop a complementary or a functionally equivalent substitute program.[291] The principal argument for allowing unlicensed second comers to use interface information to develop complementary products, but not functional equivalents (in the Commission's terms, one-way cf. two-way interoperability) is that there seems to be a much greater risk that the interface's developer will not be able to recoup its R&D expenses because functional equivalents are likely to undercut the sales of the interface developers' principal products (e.g., the platform), whereas complements are likely to build demand for the platform.[292]

This was the main concern that Microsoft raised about being forced to supply highly detailed interface information to Sun Microsystems in the Commission's competition case. Establishing a reasonable royalty for licensing interface information as to developers of complementary products may be easier than establishing one as to developers of functionally equivalent products, as the platform's developer very likely already has a licensing program for development of complementary products.

---

[287] If interfaces are more open, they then tend to be more programmer-friendly because non-proprietary programmers who implement these interfaces may refine the specification to offer more precise substitutes for opaque or confusing terms. See, e.g., Eric Wilde, *What Are You Talking About?*, IEEE International Conference on Services Computing 256 (SCC 2007).

[288] See, e.g., William H. Page & Seldon J. Childers, *Software Development as an Antitrust Remedy: Lessons From the Enforcement of the* Microsoft *Communications Protocol Licensing Requirement*, 14 Mich. Telecom. & Tech. L.Rev. 77, 130 (2007).

[289] Software Directive, supra note 3, art. 9.1.

[290] See supra notes xx and accompanying text.

[291] U.S. courts have treated interface information as unprotectable elements of programs without regard to whether the defendants' products were functional equivalents and complements. See, e.g., *Altai*, 982 F.2d at 693 (Altai's program was a direct competitor of CA's program); *Sega v. Accolade*, 977 F.2d at 1510 (Accolade's program was a complement to the Sega platform).

[292] See, e.g., Philip J. Weiser, *The Internet, Innovation, and Intellectual Property Policy*, 103 Colum. L. Rev. 534 (2003) (arguing for different IPR treatment of complements and substitutes).

Yet, some factors cut in favor of treating complements and equivalents the same. For one thing, in today's complex networked world, it is no longer as easy to distinguish between complements and substitutes. Interfaces today may enable more complex exchanges than before (e.g., at one time, a network-based program may be acting in a complementary fashion and in another, it may be acting as a substitute).

Secondly, much of the information that is necessary to make a complementary product is also necessary to know to make a functional equivalent, and vice versa. It is uncommon for IP law to treat information as protectable for one purpose, but not for another.[293] Neither copyright nor trade secrecy law has made a distinction between complements or substitutes when analyzing the lawfulness of reverse engineering and reuse of information discerned from reverse engineering.[294] Reverse engineers frequently aim to make competing and not merely complementary products.

Third, firms sometimes adopt business strategies that do not conform to the usual platform/complement story. Nintendo, for example, lost money on sales of its NES game consoles. Its strategy for financial success depended upon controlling the market for games for this platform. By reverse engineering Nintendo's interfaces and making unlicensed games for the console, Atari Games was thwarting Nintendo's recoupment strategy. Nintendo asserted its patent to stop Atari Games from selling unlicensed games because Nintendo was reaping high profits by selling its own games for the Genesis console and getting royalties from licensed developers.[295]

Finally, requiring dominant firms to provide interface information to developers of functionally equivalent products has not in the past undermined the ability of the maker of the dominant platform to recoup its investments in R&D. IBM and Fujitsu, for instance, were able to coexist in supplying functionally equivalent software for mainframe computers, both before the arbitration discussed above and afterwards. Some competition among platforms may be beneficial to consumers who at the same time also benefit from the platform's stability as a de facto standard.

It is, however, noteworthy that the three most substantial disputes over program interfaces—the European Commission's case against IBM in the 1980's, the IBM-Fujitsu arbitration, and the Commission's more recent case against Microsoft—involved efforts

---

[293] International News Service v. Associated Press, 248 U.S. 215 (1918) is a rare example in which certain information (news) was treated as "quasi-property," that is, common property as to the general public, but private property as between INS and AP. However, the *INS* concept of quasi-property has been discredited over time. See, e.g., Pamela Samuelson, *Information as Property: Do* Ruckelshaus *and* Carpenter *Signal a Changing Direction in the Law?*, 38 Cath. U. L. Rev. 365 (1989).

[294] See, e.g., Sega Enter. Ltd. v. Accolade, Inc., 975 F.2d 1510 (9th Cir. 1992)(reverse engineering to make complement); Sony Computer Ent'm't, Inc. v. Connectix Corp., 203 F.3d 596 (9th Cir. 2000)(reverse engineering to make functional equivalent). Should it really matter whether Altai reverse engineered CA's software with which it competed or the IBM programs with which it was designed to interoperate to extract interface information?

[295] See, e.g., Samuelson & Scotchmer, supra note xx, at 1619.

by dominant firms to thwart competitors from making functional equivalent products rather than complements.[296]

What we can say with some confidence is that interface patents pose the gravest risks for competition and follow-on innovation when the exercise of such patents are essential to interoperability, when the patents are held by firms with market power, and when there are incentives for firms in dominant positions to exercise their interface patents in a manner that effectively excludes competitors from the market or provides the opportunity for leveraging a dominant firm's power in one market into that of an adjacent market, especially as to disruptive new entrants with an entrepreneurial bent. Any need for regulation of program interface patents should be focused on these circumstances, rather than on interface patents as such.

[296] While the Commission is surely right that it is a common industry practice to license interface information, this practice is less common as to firms that want to make functional equivalents.