Minnesota Law Review

2009

Are Patents on Interfaces Impeding Interoperability?

Pamela Samuelson

Article

Are Patents on Interfaces Impeding Interoperability?

Pamela Samuelson[†]

Interoperability among information and communications technologies (ICT) is widely believed to promote socially desirable goals, such as fostering competition and innovation, enhancing consumer satisfaction, and promoting economic

[†] 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, Peter Lee, Mark Lemley, Suzanne Scotchmer, Carl Shapiro, Ted Sichelman, Polk Wagner, Phil Weiser, Christopher Yoo, and several software industry executives and lawyers for their useful comments on earlier drafts of the Article. Copyright © 2009 by Pamela Samuelson.

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 boundaries between ICT systems, that is, the interfaces between them.[3] Insofar as patents are issuing on interface designs and components, many worry that they will be used to impede interoperability.[4] This Article considers whether such patents are, in fact, impeding interoperability, and if so, what should be done about it.

Part I explains the significance of interfaces and interoperability, various technical challenges involved in designing and implementing interfaces, and the competitive importance of interoperability to various stakeholders. It also considers how intellectual property (IP) rules and IP strategies about interface protection evolved over time and why patents eventually emerged as an important source of IP protection. The factors that motivate firms to seek patents for ICT interfaces (or not) are complex and dynamic. Part I gives examples of interface patents that have impacted interoperability.

Part II discusses the extensive array of policy options that commentators, policymakers, and courts have considered 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.[5] Subpart B discusses ways in which patent rules might be tailored to facilitate

[1] *See, e.g.*, URS GASSER & JOHN PALFREY, BREAKING DOWN DIGITAL BARRIERS: WHEN AND HOW ICT INTEROPERABILITY DRIVES INNOVATION 1 (2007), http://cyber.law.harvard.edu/interop/pdfs/interop-breaking-barriers.pdf. Gasser and Palfrey observe that it is very difficult to find anyone who speaks out against interoperability. *Id.* It bears mentioning, however, that interoperability also enables viruses, worms, Trojan horses, spam, and other malware. *Id.* at 2.

[2] *Id.* at 4.

[3] *Cf. id.* at 9 (explaining that the simplest approach to achieving technical interoperability is through "unilateral openness").

[4] *See, e.g.*, INFORMATION SERVICES INDUSTRY DIVISION, INTERIM REPORT OF THE "STUDY GROUP ON THE LEGAL PROTECTION OF SOFTWARE AND PROMOTION OF INNOVATION" (Oct. 11, 2005), http://www.meti.go.jp/english/information/downloadfiles/051017LegalProtectionSoftware.pdf [hereinafter ISID INTERIM REPORT] (expressing concern about the exclusionary potency of patents that can impede innovation in the software sector).

[5] 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, On the Legal Protection of Computer Programs, 1991 O.J. (L 122) 42 (EC) [hereinafter Software Directive].

interoperability. Subpart C explores some proposals to subject interface patents to liability rules, such as awarding damages instead of injunctive relief as to those who use patented interfaces to achieve interoperability. Subpart D considers antitrust and 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 reviews the European Commission's order requiring Microsoft to prepare documents providing extensive technical detail about Windows interfaces and to make this information available on reasonable licensing terms to competitors in the work group server operating system (WGS-OS) market.[6] Subpart E focuses on some private sector initiatives, including those 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 nondiscriminatory (RAND) terms.

Part III observes that there is a considerable amount of interoperability in today's ICT environment, notwithstanding the issuance of many interface patents and the dire predictions of social harm that underlie proposals for strong regulatory responses to interface patents. There are generally adequate incentives for firms to make interface information available or license interface patents on reasonable terms. In view of this, it does not seem necessary or appropriate to adopt strong measures, such as excluding interfaces from patent protection or immunizing use of interface patents to achieve interoperability. Interface patents pose the gravest risks for competition and follow-on innovation when exercise 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. If regulatory intervention of interface patents is appropriate at all, it should only be undertaken in a targeted manner to address specific harms.

[6] *See* Commission of the European Communities, Commission decision of March 24, 2004, relating to a proceeding under Article 82 of the EC Treaty (Case COMP/C-3/37.792 Microsoft), Brussels, April 21, 2004, C(2004)900 final, *available at* http://ec.europa.eu/competition/antitrust/cases/decisions/37792/en.pdf [hereinafter Commission Decision]. The Commission's order was affirmed in Case T-201/04, Microsoft Corp. v. Comm'n, 2007 E.C.R. II-3601.

Yet, this Article also recommends some tailoring of patent rules in a manner that is likely to foster greater ICT interoperability without undermining incentives to invest in development of innovative interfaces. This includes closer scrutiny of patent applications that cover interface techniques, more rigorous disclosure of enablement of interface inventions, more meaningful and cost-effective post-grant review of patent validity, and judicial rulings that antireverse engineering clauses in mass-market licenses should not be enforceable.

I. THE ROLE OF IP LAW IN THE PROTECTION OF INTERFACES

Intellectual property law often plays an important role in regulating the use of ICT interfaces, but other factors also affect interoperability. To set the stage for contextualizing the role of IP law in facilitating or blocking interoperability, Section A first offers some refined definitions of interfaces, interoperability, and related concepts, and then discusses incentives that firms have to disclose interface information to enable interoperability and to assert IP rights in these interfaces. Some firms are relatively open and nonproprietary toward their interfaces, often in hopes of generating network effects for their systems. Other firms adopt business strategies that depend on withholding interface information from those who want it and using IP rights to block interoperability. Sometimes, firms' incentives toward open or proprietary interfaces change over time, as their business strategies evolve. Section B explains why makers of early ICT systems tended to be nonproprietary toward interfaces, why firms later came to rely on trade secrecy and copyright law to protect interfaces, why efforts to assert copyright protection in interfaces failed, and why this failure contributed to a rise in the role of patents in protecting interfaces. Section C considers some advantages that patents have over trade secrecy in protecting interfaces, particularly in controlling interoperability. Section D shows that interface patents are sometimes used to challenge or thwart the development of interoperable technologies.

A. SOME DEFINITIONS AND PRELIMINARY OBSERVATIONS

As applied to ICT systems, interoperability means "[t]he capability 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."[7] A recent book points out:

> Interoperability doesn't require that two systems be identical in design or implementation, only that they can exchange information and use the information 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.[8]

Interoperability can occur when the maker of one ICT system develops interfaces that enable the exchange of information between the entity it is developing and the entities with which its entity will interact, including importantly those developed by other firms.[9] The modular structure of modern ICT products, which interconnect through interfaces, has brought about more and more rapid innovation than earlier monolithic systems (e.g., mainframe computers of yesteryear) provided,

[7] Int'l Electrotechnical Comm. [IEC], *Information Technology-Vocabulary*, at 6, ISO/IEC 2382-1 (3d ed. 1993). Although this Article will concentrate on interoperability in the computing and information industries, interoperability is also a very important concept in the telecommunications industry. To enable competition in telecom markets, it is necessary for incumbents to provide resources for others to interconnect with their systems, which monopoly providers are sometimes reluctant to do. There is a long history of regulation of interconnection by both antitrust law and the Federal Communications Commission and similar agencies in other countries. *See, e.g.*, Joseph Farrell & Philip J. Weiser, *Modularity, Vertical Integration, and Open Access Policies: Toward a Convergence of Antitrust and Regulation in the Internet Age*, 17 HARV. J.L. & TECH. 85, 93–95 (2003) (discussing regulation of competition in telecommunications); *see also* 47 U.S.C. § 251(c) (2006) (discussing the duties imposed on incumbent local exchange carriers).

[8] ROBERT J. GLUSHKO & TIM MCGRATH, DOCUMENT ENGINEERING: ANALYZING AND DESIGNING DOCUMENTS FOR BUSINESS INFORMATICS & WEB SERVICES § 6.0 (2005).

[9] ICT interfaces are informational equivalents of the standard plug and socket designs that designers of electric 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* (Berkeley Center for Law and Technology, Law and Technology Scholarship, Paper No. 59, 2008), *available at* http://repositories.cdlib.org/bclt/lts/59/ (using this analogy). Although all countries have adopted a standardized electrical socket and plug design, many have adopted different socket and plug designs from one another. This is why international business travelers have to bring transformers and multiple plug kits with them when they travel. Spare parts for machinery raise similar issues to the interface issues discussed in this Article. *See* Jens Schovsbo, *As If Made for Each Other: Intellectual Property Rights and Protection of Compatible Products*, 29 INT'L. REV. OF INDUS. PROP. & COMPETITION L. 510, 525–31 (1998) (providing an argument critical of design protection legislation insofar as it impedes competition in product aftermarkets).

leading to improved components (such as memory, disk drives, and modems), mix and match experimentation to discover best functionalities, and a proliferation of applications.[10] Consumers have benefitted by this innovation and also by the competition among providers of these components, which has pushed prices down and offered more choices than monolithic systems provided.[11]

Interoperability enables modern ICT systems to be very powerful because they can call upon other systems' functionalities.[12] 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.[13] A developer that wants others to build applications for its product—which it often will in order to make the product attractive to customers—must make application programming interfaces (APIs) available to other firms.[14] APIs disclose the "hooks" (that is, standard formats for requests) that other developers must use to call upon the platform's services to carry out specific tasks for their applications.[15] 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.[16]

Programmers can generally design the internal structure of programs to implement interfaces and encode those designs in source code in many different ways.[17] The interfaces do,

[10] *See* Farrell & Weiser, *supra* note 7, at 92–93 (explaining briefly the evolution of the computer industry, including the innovation of computer components).

[11] *See id.* at 95–96 (discussing the benefits of modular systems, including innovation and lower prices).

[12] *See* David Orenstein, *Application Programming Interface*, COMPUTERWORLD, Jan. 10, 2000, at 66 (explaining that "virtually all software has to request other software to do some things for it," which occurs through application programming interfaces).

[13] *Id.*

[14] *See id.* ("Corporate developers should consider including APIs in applications they develop, especially if they expect the applications to last and interact with other applications.").

[15] *See id.*

[16] *See id.*

[17] In rare instances, functionalities may be defined to be "bit exact," that is, the interface must be encoded exactly in the same manner for the imple-

however, constrain program design to some degree because of the need to precisely conform a request for a program's services to the interface specification for those services.[18]

While APIs typically specify how applications should exchange information with the platform on which it is built, protocols are components of interfaces that often facilitate communications (i.e., interoperability) across different computers or ICT systems.[19] 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.[20] ICT systems typically involve multiple layers of functional units that interact with one another through a set of defined protocols.[21] File format specifications, including compression algorithms and digital rights management architectures, are also interfaces that affect interoperability among data files and programs and devices utilizing those files.

Another important distinction is that between interfaces and standards. Sometimes interfaces are collaboratively developed by technologists with the intent to promulgate them as standards (e.g., the Hypertext Transfer Protocol a/k/a HTTP). Other interface designs initially developed by private parties (e.g., the Portable Document Format a/k/a PDF) are later for-

mentation to be interoperable. *See* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1515–16 (9th Cir. 1992) (explaining that exact copying of certain code segments is necessary to achieve interoperability).

[18] *See, e.g.*, JONATHAN BAND & MASANOBU KATOH, INTERFACES ON TRIAL 8 (1995) (explaining that new applications have to conform to all the interface specifications of an existing operating system used by the application and the computer).

[19] *See* GLUSHKO & MCGRATH, *supra* note 8, § 9.5 (discussing protocols in the business process context). For a protocol to work across two computers, the code to implement it needs to be on both computers. With APIs, the code to implement the service only needs to be on one computer.

[20] *See id.*

[21] 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 e-mail server, for instance, can and should signal receipt of a message from another e-mail server, but not from other applications in different layers of the stack. *Id.* A good implementation of a protocol should be "robust," that is, it should be conservative in what it sends and liberal in what it receives. *See, e.g.*, INFO. SCIENCES INST., DOD STANDARD INTERNET PROTOCOL 21 (1980), http://www.faqs.org/ftp/rfc/pdf/rfc760.txt.pdf ("That is, it should be careful to send well-formed datagrams, but should accept any datagram that it can interpret (e.g., not object to technical errors where the meaning is still clear).").

mally adopted as standards. Privately developed interface designs can also become de facto standards when the platforms for which they were designed become successful in the marketplace. Some interfaces and interface techniques, however, never become standards.

Although it is useful to conceptualize ICT interoperability at a general level, it is important to realize that interoperability has somewhat different meanings in different ICT contexts.[22] 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.[23] In the context of digital identification (ID) systems, interoperability means the ability of two or more systems to understand ID information and metadata about users so that user data can be successfully transferred and exchanged among services using the ID system.[24] In the context of technically protected digital music, interoperability means that two or more players are either using the same technical protection measures or each can convert its encoding of the work into the encoding scheme of the other system so that music can be played on many devices and/or made available in a variety of online channels.[25] 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.[26] Although this Article will mainly focus on

[22] *See, e.g.*, GASSER & PALFREY, *supra* note 1, at 4–5 (explaining that the definition of interoperability depends on the views of the stakeholder).

[23] *See* Software Directive, *supra* note 5, at 43 ("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 software and hardware 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. *Id.*

[24] GASSER & PALFREY, *supra* note 1, at 5; *see also* JOHN PALFREY & URS GASSER, CASE STUDY: DIGITAL IDENTITY INTEROPERABILITY AND *E*INNOVATION 10–11 (2007), http://cyber.law.harvard.edu/interop/pdfs/interop-digital-id.pdf (providing a definition for ID information interoperability).

[25] *See* GASSER & PALFREY, *supra* note 1, at 5–6 (using a case study to provide an example of DRM interoperability); *see also* URS GASSER & JOHN PALFREY, CASE STUDY: DRM-PROTECTED MUSIC INTEROPERABILITY AND *E*INNOVATION 4–7 (2007), http://cyber.law.harvard.edu/interop/pdfs/interop-drm-music.pdf (providing a definition for DRM interoperability).

[26] *See* GLUSHKO & MCGRATH, *supra* note 8, § 9.6.1. It is also important to distinguish between the technical capability for interoperability and actual implementations of this technical capability, which are likely to depend 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.[27] Developers of platforms have a very big stake in interoperability because they generally benefit from the development of applications that work on their platforms.[28] 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.[29] Developers of complementary products benefit by interoperability because they can create products that work on platforms with large customer bases, and complements typically add value to the network.[30] Consumers benefit from interoperability insofar as they can use the same information resources on multiple platforms in a "plug and play" fashion.[31] 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. The very success of a platform may cause other developers to try to develop a platform compatible with the successful platform, arguing that consumers will benefit by having more than one source and stressing the importance of a level playing field on which competition can occur based on price, quality, and differ-

business agreements among the relevant firms. There are also important power dimensions to interoperability. Walmart, for instance, may demand that any firms that want to do business with it conform their business documents to the electronic data interface specifications that it has set forth. Other firms may be more willing to build interfaces and services that meet their suppliers' needs.

[27] *Id.*

[28] *Cf. id.* (discussing the views of stakeholders, including platform providers).

[29] *See, e.g.*, Joseph Farrell, *Standardization and Intellectual Property*, 30 JURIMETRICS J. 35, 36 (1989); Mark A. Lemley & David McGowan, *Legal Implications of Network Economic Effects*, 86 CAL. L. REV. 479, 488–95 (1998).

[30] *See, e.g.*, Philip J. Weiser, *The Internet, Innovation, and Intellectual Property Policy*, 103 COLUM. L. REV. 534, 561 (2003) (explaining the benefits America Online receives by allowing complementary products to operate on its platform).

[31] *See, e.g.*, Mark A. Lemley & David W. O'Brien, *Encouraging Software Reuse*, 49 STAN. L. REV. 255, 289 (1997). The deployment of noninteroperable systems also puts consumer investments at risk, as it may be difficult to predict which technology is likely to become the de facto standard. The market for high definition DVDs, for example, did not develop rapidly because of the noninteroperability of the Blu-Ray and HD-DVD formats.

ences in feature sets.[32] The market may become larger for all players when there is one interface and many implementations, rather than multiple platforms, each of which is noninteroperable with the other. The interests of successful platform developers and emerging competitors are, however, not always aligned.[33]

Interoperability is often conceived as a binary concept: one ICT entity either interoperates with another ICT entity or it doesn't.[34] From the users' standpoint, there is certainly something to this. But interoperability can also be conceived as a continuum or spectrum, along which some entities (e.g., programs or content) are more interoperable than others. (Market forces often require firms to be near the fully interoperable end of the continuum; in most cases, products that are only 90 percent interoperable will find it difficult to compete against those that are more fully interoperable.) 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 details of its design, 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.[35] Microsoft does not, however, disclose all of the interface information that ISVs might want to know. Often the undisclosed information pertains to how one part of its system calls upon the services of another component.[36] ISVs may want greater access to information about these system calls to achieve better per-

[32] *See, e.g.*, BAND & KATOH, *supra* note 18, at 330–34

[33] *Id.*

[34] *See* GASSER & PALFREY, *supra* note 1, at 4 (providing a definition of interoperability).

[35] *See* MSDN Library, http://msdn.microsoft.com/en-us/library/default.aspx (last visited Apr. 29, 2009) (containing many of Microsoft's APIs).

[36] Although Microsoft may justify nondisclosure 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.*, Microsoft's Allegedly Undocumented APIs—*Comes v. Microsoft*, http://www.groklaw.net/articlebasic.php?story=2007020819534335 (Apr. 29, 2007) (providing the expert's supplemental report in *Comes v. Microsoft* explaining Microsoft's use of undisclosed information to gain advantages over other firms).

formance or build a richer feature set for their programs.[37] Frustrated about Microsoft's limited disclosure of internal APIs, some have engaged in reverse engineering to discern and document unlicensed interface information.[38]

Microsoft is far from the only firm that has adopted a business strategy that depends on controlling access to interface information and interoperability. Firms may adopt a controlled interoperability strategy because it may be more lucrative than a completely open, fully interoperable strategy. Consumers are presumably willing to pay a premium for value they perceive they are getting from at least some noninteroperable products. 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.[39] 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.[40] 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.[41] To compete effectively against Apple, RealNetworks has incentives to develop technology and music services that would be more attractive to consumers.[42] Competition among proprietary systems, as well as among open systems, can promote social welfare.[43]

Microsoft and Apple are among the many firms that rely on IP rights as one means by which to control interoperability and

[37] Whether undocumented details are "essential" to achieving interoperability can be a matter of some debate. *See, e.g.*, Nokia Corp. v. Interdigital Tech. Corp., [2007] EWHC (Pat) 445, [1]-[35] (Eng.) (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).

[38] Numerous books have disclosed such APIs. *See, e.g.*, SVEN SCHREIBER, UNDOCUMENTED WINDOWS 2000 SECRETS (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.

[39] GASSER & PALFREY, *supra* note 1, at 6.

[40] *See id.* at 9.

[41] *Cf. id.* (expressing "objections" over the reverse engineering).

[42] *See id.* at 9–10.

[43] *Cf.* Weiser, *supra* note 30, at 536–40 (discussing the benefits of open systems and competitive platform models on society).

to stop unlicensed persons from offering compatible technologies. Because IP rights have played a complex and dynamic role in regulating interoperability, Section B will explain how IP law evolved as to interfaces and why patents were slow in becoming an important form of protection for interfaces.

B. EVOLVING IP RULES AND PRACTICES AS TO INTERFACES

In the early years of the industry, computers were standalone devices that were typically physically connected to only a few terminals and output devices. Anyone who wanted to process the same data or programs on different machines had to hand-carry the punch cards or magnetic tape on which the information was stored from one machine to the next. After the development of computers capable of storing and executing programs and processing data, it became evident that customers valued having compatible systems. To aid customers in achieving compatibility, many firms, including IBM, often distributed source code and interface specifications without IP restrictions.[44] 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.[45]

[44] *See, e.g.*, BAND & KATOH, *supra* note 18, at 19; *see also* Anita Stork, Note, *The Use of Arbitration in Copyright Disputes*: IBM v. Fujitsu, 3 HIGH TECH. L.J. 241, 242–43 (1987) (pointing out that IBM distributed source code without copyright restrictions through the 1970s).

[45] The open-publication strategy of that era was also possibly 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” (meaning that there is an express understanding that doubt exists as to whether the code qualifies for copyright protection should litigation later occur). Indeed, the registration certificates indicated the Office’s doubt about the copyrightability of programs in machine-executable form. *See, e.g.*, COPYRIGHT OFFICE, CIRCULAR 31D (Jan. 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-Readable Form*, 1984 DUKE L.J. 663 (explaining that the Final Report of the National Commission on New Technological Uses of Copyrighted Works (CONTU) failed to consider the primary aims of copyright

Starting in the mid- to late-1970s, manufacturers of computer systems came to realize that interfaces were commercially valuable, and it became more common for firms to withhold source code or interface information from those who wanted access to them.[46] Firms began to think of source code and interface specifications as trade secrets and to distribute programs in object code form.[47] They claimed copyright protection for that code and often hoped copyright would also protect them against reverse engineering (which inevitably requires making intermediate copies of the code), thus indirectly preventing trade secrets in interfaces from being discerned and thwarting the efforts of unlicensed parties to make interoperable systems.[48]

In the mid-1980s to the mid-1990s, some firms, IBM prominently among them, also argued copyright protection should be available for original interfaces embodied in programs.[49] The issue first arose in litigation when the manufacturer of an Apple clone computer claimed 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.[50]

and patent law when recommending that current intellectual property law protect machine readable computer programs). 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, 71–72 (1972) (denying patentability of an 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) [hereinafter Samuelson, Benson *Revisited*] (discussing case law and doctrinal developments related to the patentability of program algorithms and program-related inventions).

[46] *See, e.g.*, Farrell & Weiser, *supra* note 7, at 92 ("To keep its system closed, IBM kept the interfaces between the different parts of its system secret and proprietary.").

[47] Allen R. Grogan, *Decompilation and Disassembly: Undoing Software Protection*, COMPUTER LAW., Feb. 1984, at 1.

[48] *See id.* (discussing the core argument for this approach).

[49] Trademark and false advertising law is sometimes used to challenge those who develop compatible ICT systems. *See, e.g.*, Creative Labs, Inc. v. Cyrix Corp., 42 U.S.P.Q.2d 1872, 1876 (N.D. Cal. 1997) (granting a preliminary injunction against the sale of sound cards that were not "fully compatible" with the plaintiffs' system, as the defendant's ads claimed).

[50] *See* Apple Computer, Inc. v. Franklin Computer Corp., 714 F.2d 1240, 1245 (3d Cir. 1983). Franklin also argued that Congress had only intended to protect application programs that interacted with people, not purely functional programs such as operating systems. *See id.* at 1246–52. The court did not find this or other Franklin defenses persuasive. *See id.* at 1251. This challenge was

The court responded:

> 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.[51]

This dicta dimmed the prospects for success of future compatibility defenses to copyright claims.

Further dimming these prospects was the Third Circuit's decision in *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*[52] *Whelan* characterized computer programs as "literary works" and reasoned that since copyright law had long protected nonliteral 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.[53] *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.[54] 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 *Whelan* court worried that there would be too little protection to provide proper incentives to develop computer programs.[55]

Because interfaces are unquestionably part of program SSO, *Whelan* seemed to extend copyright protection to them. When Computer Associates (CA) sued Altai for copying of CA's interfaces,[56] it relied heavily on *Whelan*.[57] CA pointed to sub-

somewhat surprising, given that Congress had amended copyright law to clarify that programs could be copyrighted. Act of Dec. 12, 1980, Pub. L. No. 96-517, 94 Stat. 3015, 3028 (codified at 17 U.S.C. §§ 101, 117 (2006)).

[51] *Franklin*, 714 F.2d at 1253.

[52] 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 Pamela Samuelson, *Why Copyright Law Excludes Systems and Processes From the Scope of Its Protection*, 85 TEX. L. REV. 1921, 1962–73 (2007).

[53] *Whelan*, 797 F.2d at 1233–40, 1248.

[54] *Id.* at 1236. 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." *Id.* at 1228, 1247.

[55] *Id.* at 1237.

[56] Computer Assocs. Int'l, Inc. v. Altai, Inc., 775 F. Supp. 544 (E.D.N.Y. 1991), *aff'd in part, vacated in part*, 982 F.2d 693 (2d Cir. 1992).

[57] *See* Reply Brief for Plaintiff-Appellant *passim*, Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992) (No. 91-7893) (relying heavily on *Whelan*).

stantial 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).[58] CA argued the parameter lists had been carefully and precisely designed, making them costly to develop and commercially significant parts of its programs.[59] 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.[60]

Altai was able to persuade the court to recognize that external factors sometimes constrain program design choices.[61] Because CA-Scheduler and Oscar provided the same scheduling services and both were designed to interoperate with the same IBM OS programs, similarities in their parameter lists were understandable and not evidence of infringement.[62] The court in *Altai* asserted that extending copyright protection to program interfaces would "have a corrosive effect on certain fundamental tenets of copyright doctrine"[63] and suggesting patents as an alternative form of IP protection for interfaces.[64]

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

[58] *Id.* at 5–6.

[59] *See id.* at 3.

[60] *See id.* at 14 (discussing some of the goals in seeking copyright protection).

[61] *See* BAND & KATOH, *supra* note 18, at 9. The court in *Altai* 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* Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 709–10 (2d Cir. 1992) (citing 3 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 13.01, at 13-4 (1991)).

[62] *See* Brief of Defendant-Appellee at 12–14, Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992) (No.91-7893).

[63] *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. Protecting interfaces would be corrosive of copyright principles because they are too functional to be protected as original expression of an author. *See, e.g.*, Samuelson, Benson *Revisited*, *supra* note 45, at 1146 n.472 (explaining that copyright and patent law should be "separate and exclusive"); *infra* Parts II–III.

[64] *Altai*, 982 F.2d at 712.

emerge as the leading decision about software copyrights.[65] However, the patent option became more attractive after the Ninth Circuit's decision in *Sega Enterprises, Ltd. v. Accolade, Inc.*[66]

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.[67] Second, *Sega* followed *Altai* in ruling that program interfaces were elements of programs that copyright law did not protect; indeed, *Sega* spoke of interface information as "functional requirements for achieving compatibility with other programs."[68] Third, the court ruled that copying program code in the course of reverse engineering for a legitimate purpose such as extracting interface information to make a compatible program did not infringe copyright.[69] 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.[70]

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.[71]

[65] *Altai* has been followed in at least forty-nine subsequent decisions. *See, e.g.*, Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807 814–15 (1st Cir. 1995). Lotus relied heavily on *Whelan* in support of its claim that Borland infringed its copyright in the command hierarchy of the Lotus 1-2-3 user interface. The First Circuit ruled that the command hierarchy was an unprotectable method of operating a spreadsheet program under 17 U.S.C. § 102(b). *Id.* at 815. Although the Supreme Court took Lotus's appeal from the First Circuit's ruling, the Court was evenly divided 4-4 on the merits, and hence affirmed the lower court's ruling. *See* Lotus Dev. Corp. v. Borland Int'l, Inc., 516 U.S. 233, 233 (1996) (per curiam). By 1996, however, *Altai* had become the standard software copyright-infringement case, displacing *Whelan*.

[66] 977 F.2d 1510 (9th Cir. 1992).

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

[68] *Id.* at 1525–26.

[69] *Id.* at 1527–28 (finding reverse engineering copies to be fair use).

[70] *Id.* at 1526.

[71] *See id.* at 1516, 1528–32 (treating certain Sega code as too functional for copyright protection).

After *Sega*, developers could no longer hope to protect interfaces by copyright. Because *Sega* endorsed unlicensed copying of code to extract interface information,[72] it imperiled developer efforts to protect interfaces as trade secrets.[73] *Sega* sigsignaled that the only reliable means for protecting the functional requirements for achieving interoperability was by patenting them. Patents have a key advantage over copyrights in protecting interfaces because patent law has no "merger" doctrine.[74] 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.[75] Moreover, patent law also has no explicit reverse engineering privilege.[76]

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 the courts became more receptive to software patents.[77] The Supreme Court had initially cast doubt

[72] Prior to *Sega*, some commentators had argued that reverse engineering of object code should be treated as both copyright infringement and trade-secret misappropriation. *See e.g.*, Grogan, *supra* note 53, at 4–5. Some commentators argue that reverse engineering should be treated as copyright infringement because of the intermediate copying required to reverse engineer and as 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. *Id.*

[73] *See* Victoria A. Cundiff, *Trade Secrets and the Internet: Preventing the Internet from Being an Instrument of Destruction*, *in* 12TH ANNUAL INSTITUTE ON INTELLECTUAL PROPERTY LAW 403, 413 (PLI Intellectual Prop. Course Handbook Series No. G-877, 2006), WL 877 PLI/PAT. 403.

[74] *Cf.* Oskar Liivak, *The Forgotten Originality Requirement: A Constitutional Hurdle for Gene Patents*, 87 J. PAT. & TRADEMARK OFF. SOC'Y 261, 296-97 (2005) (advocating for an adoption of the merger doctrine for patents).

[75] Patent law has, however, a number of policy options 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 *passim* (2003).

[76] *But see generally* Julie E. Cohen et al., *Patent Scope and Innovation in the Software Industry,* 89 CAL. L. REV. 1 (2001) (exploring patent doctrines that might permit reverse engineering of software even if some components of the software were patented).

[77] The European experience with software patents and special concerns about interface patents are discussed below in Part II.A. 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. *See Commission Proposal for a Directive of the European Parliament and of the Council on the Patentability of Computer-Implemented Inventions*, at 2–6, COM (2002) 92 final (Feb. 20, 2002), *available at* http://www2.europarl.eu.int/oeil/file.jsp?id=219592 [hereinafter *Proposed Software Patent Directive*]. *See generally* REINIER BAKELS & P. BERNT HUGENHOLTZ, THE PATENTABILITY OF COMPUTER PROGRAMS: DISCUSSION OF EUROPEAN-LEVEL LEGISLATION IN

on the patentability of program innovations in the 1970s,[78] but the Court was receptive to one such patent in its 5-4 decision in 1981 in *Diamond v. Diehr*.[79] Relying on some broad language in *Diehr*,[80] the Federal Circuit during the 1980s and 1990s developed a capacious conception of patentable subject matter under which virtually all computer program-related inventions are patentable.[81] This, coupled with increasing "thinness" of copyright protection after *Altai* and *Sega* achieved widespread acceptance in the mid-1990s, led to a big surge in the patenting of software innovations,[82] including the issuance of patents on interfaces.

Although program interfaces can generally be patented in the United States,[83] firms do not always choose to seek patents

THE FIELD OF PATENTS FOR SOFTWARE (2002) (discussing European-level legislation in the field of patents for software), *available at* http://www.ivir.nl/publications/other/softwarepatent.html.

[78] *See* Parker v. Flook, 437 U.S. 584, 584 (1978); Gottshalk v. Benson, 409 U.S. 63, 71–73 (1972).

[79] 450 U.S. 175, 175–76 (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. *Id.* at 177. The PTO rejected Diehr's claim because its only novelty lay in the program's calculations. *Id.* at 179–81. 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, the majority opinion did not repudiate the Court's earlier rulings on the unpatentability of certain program innovations, and *Diehr* involved a traditional manufacturing process (i.e., curing rubber). *Id.*

[80] *Diehr*, 450 U.S. at 182 (stating that patentable subject matter includes everything "under the sun made by man") (citations omitted).

[81] *See, e.g.*, AT&T Corp. v. Excel Commc'ns, Inc., 172 F.3d 1352, 1353 (Fed. Cir. 1999) (holding that the claimed subject matter was patentable).

[82] *See* Josh Lerner et al., *What is the Impact of Software Patent Shifts?: Evidence from* Lotus v. Borland 10 (Nat'l Bureau of Econ. Research, Working Paper No. 11168, 2005), *available at* http://www.nber.org/papers/w11168 (noting that there is empirical evidence of a surge in the patenting of software dating from the mid-1990s).

[83] Software industry representatives with whom I spoke estimate that there are many thousands of patents on interfaces. Lexis searches for patents using search terms such as "application program interfaces" and "communications protocols" yield thousands of "hits" (Lexis search in the MEGA database on March 5, 2009 confirms this result). Patent lawyers believe there are likely many more interface patents than these searches reveal because patent applications often do not mention the term "interface" or "communications protocol." Interfaces are sometimes claimed as a component of a feature they enable. *See* Scott Elengold, Note, *An Inquiry Into Computer System Patents: Breaking Down the "Software Engineer"*, 61 N.Y.U. ANN. SURV. AM. L. 349, 360 (2005). Most interface patents do not cover the totality of the interface, but small parts of interfaces. *Id.*

for interface designs.[84] Not patenting interfaces makes sense for developers whose business strategy relies upon publication of interfaces as a potential generator of network effects, as well as for collaboratively developed interfaces, such as those for open source projects.

Even firms with more proprietary approaches toward interfaces may still have good reasons not to patent them. For one thing, program interfaces can often be protected quite effectively as trade secrets.[85] Because commercially distributed programs are typically shipped in machine-executable 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;[86] it also obviates the need for disclosure of any innovation the interface embodies.[87]

Trade secrecy can, of course, be jeopardized by reverse engineering conducted by those who want to access interface information, but firms can and often do try to counteract this risk by inserting antireverse engineering clauses into their license agreements,[88] or by obfuscating the design to make interfaces difficult to discern. The more complex a program is, moreover, the more difficult it will be to access interfaces through reverse engineering.[89] If unlicensed parties successfully reverse engineer a firm's interface, the firm whose products have been reverse engineered can change the interface in subsequent versions, thereby thwarting interoperability by unlicensed firms.[90] Still, some interfaces are patented, so it is worth considering reasons why patenting interfaces sometimes may make sense.

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

[85] *See* Pamela Samuelson & Suzanne Scotchmer, *The Law and Economics of Reverse Engineering*, 111 YALE L.J. 1575, 1607–30 (2002).

[86] *See* Douglas A. Barnes, Note, *Deworming the Internet*, 89 TEX. L. REV. 279, 292 (2004).

[87] Trade secrecy also lessens the risk that the developer will be charged with infringing someone else's interface patent. *See id.*

[88] This is a common practice in the software industry. *See, e.g.*, Samuelson & Scotchmer, *supra* note 85, at 1626–30.

[89] *Cf.* Lemley & McGowan, *supra* note 30, at 529 ("[R]everse engineering an operating system's program to achieve compatibility in a competing program is particularly difficult.").

[90] *See id.*

C. INCENTIVES FOR PATENTING INTERFACES

Interfaces can be costly to develop and may embody significant innovations,[91] which explains why firms sometimes want to rely on patent incentives to recoup investments in these technologies. But an even more powerful reason to patent interfaces may derive from the strong blocking power such patents can confer over the development not only of competing but also of complementary products insofar as the interface defines the boundaries between ICT systems. [92]

Interface patents are also valuable to their developers because it may be impossible to work around them.[93] Even a very narrowly drawn interface patent may preclude interoperability as to key functions.[94] Detecting infringement of interface patents is generally easier than of other software patents because if unlicensed products successfully interoperate with the patentee's products, they likely infringe.[95] The exclusionary power of interface patents is, moreover, strong even if the technical design disclosed in the patent is only modestly innovative or an

[91] *See, e.g.*, William H. Page & Seldon J. Childers, *Bargaining in the Shadow of the European* Microsoft *Decision: The Microsoft-Samba Protocol License*, 102 NW. U. L. REV. COLLOQUY 332, 335, 354 (2008), *available at* http://www.law.northwestern.edu/lawreview/colloquy/2008/16/LRColl2008n16Page&Childers.pdf (characterizing Microsoft's Active Directory interfaces as innovative, distinctive, and costly).

[92] *See, e.g.*, Atari Games Corp. v. Nintendo of Am., Inc., 30 U.S.P.Q.2d (BNA) 1401, 1422 (N.D. Cal. 1993) (providing that the developer of a complementary product infringed an interface patent).

[93] *See, e.g.*, Mark A. Lemley & Carl Shapiro, *Patent Holdup and Royalty Stacking*, 85 TEX. L. REV. 1991, 2016 (2007) (discussing the holdup problem arising from an inability to design around a patent on a component of a multicomponent technology).

[94] *See, e.g.*, BAKELS & HUGENHOLTZ, *supra* note 77, at 21–22 (giving an example of reverse engineering patents). Narrowly drawn interface patents have an advantage over broadly written interface patents because narrow patents are generally easier to defend against invalidity challenges. *See id.*

[95] Patents on internal designs of programs (e.g., algorithms) are, by contrast, often difficult to enforce offensively (i.e., to stop competitors from using them) because such designs are typically difficult to discern from executing commercially distributed object code. *See, e.g.*, Gideon Parchomovsky & R. Polk Wagner, *Patent Portfolios*, 154 U. PA. L. REV. 1, 30–31 (2005). Although firms often seek patents for internal design components, patents on such innovations are generally more useful for defensive than for offensive purposes. *See id.* at 31–43. That is, developers tend to seek patents on such internal design elements to assure themselves of having the freedom to operate in developing 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 id.*

arbitrary variation on an existing technique.[96] This means that, absent contractual commitments or licensing obligations such as those that may be imposed by standard setting organizations (SSOs),[97] firms can usually charge higher royalty rates for licensing interface patents than for licensing other patents, regardless of the degree of innovation the interface patents may embody.[98] 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 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.[99] Determined reverse engineers may be motivated to discover obscure aspects of interfaces.[100] The enforceability of license restrictions on reverse engineering has, moreover, been widely questioned.[101] Although copyright law protects program code, any interfaces embedded in programs are beyond the scope of copyright's protection.[102] 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.[103] Neither the PTO nor the courts seem to require much disclosure from developers of ICT interface techniques.[104] Firms may

[96] *See, e.g.*, Maureen O'Rourke, *Toward a Doctrine of Fair Use in Patent Law*, 100 COLUM. L. REV. 1177, 1218 (2000) (noting that many interfaces are arbitrary, obvious, and of low intrinsic value).

[97] *See infra* Part II.E.

[98] *See, e.g.*, Lemley & Shapiro, *supra* note 93, at 2009 ("[T]he patent holder can demand almost five times as much money once the industry has made irreversible investments."); O'Rourke, *supra* note 96, at 1218 ("The creation of some interfaces may entail large costs and an interface may be valuable compared to others in the market.").

[99] JAMES POOLEY, TRADE SECRETS § 5.02[1] (1998).

[100] *Id.* § 4.04[4]; *see also* Samuelson & Scotchmer, *supra* note 85, at 1587 n.49–50.

[101] Samuelson & Scotchmer, *supra* note 85, at 1626–30 (reviewing the controversy over the enforceability of antireverse engineering clauses and why most scholars think such clauses should not be enforced, particularly in mass-market licenses).

[102] *See infra* notes 150–152 and accompanying text.

[103] Sega Enters., Ltd. v. Accolade, Inc., 977 F.2d 1510, 1526 (9th Cir. 1992).

[104] *See, e.g.*, *In re* Hayes Microcomputer Prods., Inc., 982 F.2d 1527, 1532–39 (Fed. Cir. 1992) (rejecting challenges to an interface patent based on the inadequacy of written description and best mode disclosure requirements of patent law); *see also* Cohen et al., *supra* note 76, at 18 ("[T]he Federal Cir-

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.[105]

Established firms are more likely than entrepreneurs to patent interfaces to gain control over the development of compatible systems. Entrepreneurs may sometimes seek patents on interfaces to attract venture capital.[106] Yet, patented interfaces owned by entrepreneurial firms are unlikely to confer substantial market power because these firms will generally need to license such patents on reasonable terms to entice others to develop products or services for their platform.[107]

Incentives to seek patents for interfaces may, of course, change over time. While a startup might be willing to disclose

cuit does not require would-be patentees of software inventions to disclose . . . very much at all about their inventions.").

[105] Software developers cannot seek patent protection for documents detailing interface specifications, as such documents would be ineligible for patent protection as "printed matter." *See, e.g.*, *In re* Russell, 48 F.2d 668, 668–69 (C.C.P.A. 1931) (holding a novel method of arranging surnames by phonetic groupings to be unpatentable subject matter). 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* Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 703 (2d Cir. 1992); *Sega*, 977 F.2d at 1524.

[106] *See, e.g.*, Ronald J. Mann, *Do Patents Facilitate Financing in the Software Industry?*, 83 TEX. L. REV. 961, 973 (2007) (arguing that software patents facilitate the financing of startup firms). Some venture capitalists, however, regard software patents as a drag on innovation in the software industry. *See* Abolish Software Patents, http://www.feld.com/wp/archives/2006/04/abolish-software-patents.html (Apr. 10, 2006).

[107] 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. *See generally* U.S. Patent No. 6,125,391 (filed Oct. 16, 1998). 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 Commerce One's platform obtained a royalty-free perpetual license to practice the invention; interface protocols and XML sample documents were available under open licenses. *See, e.g.*, John Markoff, *Auction of Internet Commerce Patents Draws Concern*, N.Y. TIMES, Nov. 16, 2004, at C4. When Commerce One went bankrupt, however, these patents were the most valuable asset that Commerce One owned. *See id.* A bankruptcy auction brought in over $15 million for them. *See* Lorie Long, *Hanging on for Dear Life*, BLUE RIDGE BUS. J., Feb. 2005, *available at* http://www.bizjournal.com/content/article.php?id=1. 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, *Secretive Buyer of Some E-Commerce Patents Turns Out to Be Novell*, N.Y. TIMES, May 5, 2005, at C3.

interfaces to attract customers and partners, when its ICT system becomes successful in the marketplace, the same firm may become increasingly proprietary about its interfaces and more inclined to seek patents for extensions of existing interfaces or for new ones.[108] Exerting proprietary control over interfaces is also more likely as growth of the platform flattens, and its developer makes fewer investments in potentially disruptive innovations and more in maintaining control over the existing market.

D. EXAMPLES OF INTERFACE PATENTS

One ICT interface patent that was exercised to block the development of interoperable technologies was Nintendo's patent for a relatively high-level design for a program-to-program interface for its Nintendo Entertainment System (NES).[109] The NES included a game console, a monitor, and controls to allow users to operate games played on the console.[110] Nintendo developed games for its platform, but also licensed some other firms to do the same.[111] Loaded onto the NES console was an initialization program called 10NES, which served as an authentication protocol so that only Nintendo's games or those licensed by Nintendo could successfully be played on the NES platform.[112] Nintendo-authorized 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.[113]

Through a combination of reverse and social engineering,[114] 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 in-

[108] Patents on interfaces may amplify the network effects noted above. *See e.g.*, BAKELS & HUGENHOLTZ, *supra* note 77, at 22.

[109] *See* U.S. Patent No. 4,799,635 (filed Dec 23, 1985).

[110] *See* Atari Games Corp. v. Nintendo of Am., Inc., 30 U.S.P.Q.2d (BNA) 1402, 1402 (N.D. Cal. 1993).

[111] *See id.* at 1403.

[112] *See id.* at 1402–03.

[113] *See id.*

[114] The social engineering occurred when an 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 id.* at 1402–03.

formation processing apparatus.[115] By patenting this authentication technique, Nintendo was able to exclude Atari Games from making compatible products for its console and obtain damages for the latter's infringing uses.[116]

Sega Enterprises, a fierce competitor of Nintendo's, licensed a patent on a Trade Mark Security System (TMSS) interface technique which it then embedded in its Genesis video game system in a similar effort to prevent unlicensed videogame developers from producing games compatible with its popular console.[117] In *Sega Enter., Ltd. v. Accolade, Inc.*, the Ninth Circuit 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."[118]

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.[119] 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 interopera-

[115] *See id.* Atari Games sought a declaratory judgment of noninfringement as to both copyright and patent infringement claims in response to a threat of litigation by Nintendo. *See* Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 835–37 (Fed. Cir. 1992). 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. *Id.* The Federal Circuit upheld Atari's 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 that might be used to thwart compatibility in the future. *See id.* at 839–40. The lower court subsequently granted summary judgment to Nintendo on patent claims. *See Atari Games*, 30 U.S.P.Q.2d at 1414.

[116] The court concluded that Atari Games was a contributory infringer, not a direct infringer, of this patent. *See Atari Games*, 30 U.S.P.Q.2d at 1414; *see also* Julie E. Cohen, *Reverse Engineering and the Rise of Electronic Vigilantism: Intellectual Property Implications of "Lock-Out" Programs*, 68 S. CAL. L. REV. 1091, 1101 (1995) (discussing both the patent and copyright claims in *Atari Games* at length).

[117] *See* U.S. Patent No. 4,462,076 (filed June 4, 1982).

[118] 977 F.2d 1510, 1515 (9th Cir. 1992).

[119] *See id.* at 1520–27.

bility with the Sega platform, the Ninth Circuit ruled there was no trademark infringement.[120] Had Sega owned the patent on TMSS, it would have been able to stop Accolade from making games for its platform.

Patents on communications protocols have had powerful exclusionary effects in other litigated cases involving widely used ICTs. One was a patent on an improved method for controlling modes of modem operations that Hayes Microcomputer used in its SmartModem products which became a de facto standard in the modem market.[121] 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 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.[122] Hayes arbitrarily designated as "+++_" as the predetermined command that 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 then affirmed the verdict and issuance of an injunction.[123]

Also successful was Verizon Services' lawsuit against Vonage over a patented interface technique.[124] Vonage began pro-

[120] *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.

[121] U.S. Patent No. 4,549,302 (filed Oct. 11, 1983). The patent and some of its claims are discussed in *In re Hayes Microcomputer Prods, Inc.*, 982 F.2d 1527, 1531–32 (Fed. Cir. 1992).

[122] *Id.*

[123] *Id.* at 1530.

[124] *See generally* Verizon Servs. Corp. v. Vonage Holdings Corp., 503 F.3d 1295 (Fed. Cir. 2007). Vonage was also sued by AT&T and Sprint Nextel for infringing patents on methods and systems related to interconnecting traditional telephone calls with Internet-protocol networks. *See* Martyn Williams, *AT&T and Vonage End Patent Feud*, PC WORLD, Dec. 23, 2007, http://www.pcworld.com/article/140800/atandt_vonage_end_patent_feud.html. Qualcomm and Nokia have just recently settled litigation over licensing terms for essential patents compliant with the standards of the European Telecommunica-

viding 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.[125] 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.[126] A jury ruled against Vonage's challenge to the patents' validity and awarded Verizon $58 million in damages.[127] The trial judge stayed injunctive relief, pending Vonage's appeal.[128] The Federal Circuit affirmed the finding of infringement as to two of the patents,[129] although remanding the case for reassessment of damages;[130] yet, it affirmed the issuance of an injunction.[131] Within the Internet telephony community, concerns arose about the implications of Verizon's patents for VoIP services more generally.[132]

Although Microsoft has not sued anyone for infringement of its 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, over sixty-five U.S. patents and more than ten European patents on work group serv-

tions Standards Institute (ETSI). *See* Tatum Anderson, *Nokia-Qualcomm Settlement Boosts Sharing, Ends Costly Patent War*, INTELLECTUAL PROPERTY WATCH, July 31, 2008, http://www.ip-watch.org/weblog/2008/07/31/nokia-qualcomm-settlement-boosts-sharing-ends-costly-patent-war/.

[125] *See Verizon*, 503 F.3d at 1298.

[126] *See id.* at 1299.

[127] *See id.* at 1301–02.

[128] *See id.* at 1301.

[129] *See id.* at 1305.

[130] *See id.* at 1310.

[131] *See id.* at 1311. Vonage relied upon the Supreme Court's decision in *eBay Inc. v. MercExchange, LLC* 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 (citing eBay Inc. v. MercExchange, LLC, 547 U.S. 388, 391 (2006) (holding that courts have discretion not to issue injunctions in patent infringement cases)). The Federal Circuit was not persuaded. *Id.* at 1311. The implications of the *eBay* case for interface patents are discussed in Part II.C. 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, Apr. 2, 2007, http://www.arstechnica.com/tech-policy/news.ars/post/2007/04/02-vonage-hangs-up-on-verizon-patent-infringement-with-new-agreement.ars.

[132] *See* Adario Strange, *The Future of Internet Telephony Could Hang on Vonage Case*, WIRED, Apr. 26, 2007, http://www.wired.com/techbiz/it/news/2007/04/vonage_appeal.

er and program protocols,[133] and it has applied for numerous patents for similar protocols.[134] Microsoft relied on some of these patents as a justification for refusing to provide and license interface information to Sun Microsystems and others in a case brought by the European Commission charging it with abuse of dominant position.[135] Microsoft also owns an interface patent on aspects of its Advanced Streaming Format (ASF).[136] Some open source programmers 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.[137]

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

In the past two decades, commentators and policy makers have proposed a remarkably varied array of policy options to deal with the exclusionary potency of patents on ICT interfaces. Although owners of interface patents would likely assert that they should be free to exercise their patent rights as they wish,[138] commentators, policymakers, and courts that analyzed

[133] *See* MICROSOFT CORP., EUROPEAN UNION AND UNITED STATES PATENTS AND PATENT APPLICATIONS FOR MICROSOFT WORK GROUP SERVER PROTOCOL PROGRAM PROTOCOLS (2008), *available at* http://download.microsoft.com/download/2/8/a/28a250e5-5b79-4547-9959-346736ed7a97/WSPP_Patent_Mapping.pdf [hereinafter MICROSOFT PATENT MAPPING].

[134] *See id.*

[135] Case T-201/04, Microsoft Corp. v. Comm'n, 2007 E.C.R. II-3601; *see also infra* Part II.D.

[136] *See, e.g.*, Posting of Andy Tai to Advogato, http://www.advogato.org/article/101.html (June 5, 2000, 22:15 UTC) ("Microsoft patents the ASF media file format.").

[137] *Id.* ("Microsoft . . . stops the author of VirtualDub, a GPLed video capture and processing program for Windows, from supporting ASF.").

[138] 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 free of any kind of regulatory scrutiny. The Federal Circuit's decisions in the *Hayes* and *Verizon* cases, discussed *supra* notes 121–131 and accompanying text, are, however, consistent with this position.

the impacts of interface patents on interoperability have overwhelmingly favored some regulation of interface patents.

Section A reviews proposals to exclude interfaces from patent protection or to immunize the exercise 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.[139] Some Sun executives believe that interfaces affecting interoperability should be free from IP restrictions and be treated as a commons upon which all comers should be free to build.[140] Alternatively, some have proposed abolishing software patents altogether, which would obviously sweep away patents on computer program interfaces.[141]

[139] *See* BAND & KATOH, *supra* note 18, at 332–34.

[140] Interview with Greg Papadopoulos, Chief Tech. Officer and Executive Vice President of Research and Dev., Sun Microsystems, in San Jose, Cal. (May 8, 2008). 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* YOCHAI BENKLER, THE WEALTH OF NETWORKS 106–07 (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 below in Part II.C, are consistent with the commons approach toward interfaces that is essential to interoperability.

[141] A coalition of nonprofit organizations affiliated with the free- and open- source software movements has formed to espouse the abolition of software patents. *See generally* End Software Patents, http://endsoftpatents.org (last visited Apr. 29, 2009) (declaring that its goal is to “restore the software market to a world run by innovators, not judges”). At least one venture capitalist agrees with the gist of this coalition’s arguments. *See* Abolish Software

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.[142] 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 to license its use, the Commission flatly denied that Microsoft owned any Intellectual Property Rights (IPRs) in the interfaces the Commission ordered it to disclose to Sun and others.[143] The interfaces, in the Commission's view, were "ideas and principles" under the Software Directive.[144]

Although the EU Software Directive is often viewed as having endorsed copyright as a means of legal protection for computer programs,[145] 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.[146] The Software Di-

Patents, *supra* note 106. There are reasons to question the patentability of software innovations as a matter of U.S. law. *See generally* Samuelson, Benson *Revisited*, *supra* note 45 (discussing various arguments against patents on computer program and other information innovations).

[142] *See* Software Directive, *supra* note 5.

[143] 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. *Cf.* Commission Decision, *supra* note 6, ¶ 190 n.249 ("[S]ince the relevant specifications are not available for scrutiny, it is not possible for the Commission to determine to what extent Microsoft s claims relating to various intellectual property rights are justified."). However, the Commission defended its order before the Court of First Instance (CFI) by asserting that Microsoft had *no* IPRs in its interfaces. *See* Case T-201/04, Microsoft Corp. v. Comm'n, 2007 E.C.R. II-3601, ¶¶ 276–78. While it is possible that the Commission may only have been questioning the validity of Microsoft's interface patents, the CFI decision points to the Software Directive's recitals, which refer to interfaces as ideas and principles. *Id.* ¶ 276. 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 Part.

[144] Software Directive, *supra* note 5, at 43; *see also* BAND & KATOH, *supra* note 18, at 83; W.R. Cornish, *Inter-operable Systems and Copyright,* 11 EUR. INTELL. PROP. REV. 391, 391 (1989) (arguing that interface specifications are unprotectable ideas or principles). *But see* Ashwin van Rooijen, *Essential Interfaces: Exploring the Software Directive's Equilibrium Between Intellectual Property Rights and Competition Law*, 5 COMPUTER L. REV. INT'L (CRi) 129, 136 (2007) ("[I]nterface specifications are not excluded from protection as such.").

[145] *See* Software Directive, *supra* note 5, at 44.

[146] I have previously argued in favor of a sui generis form of legal protection for programs. *See* Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308 *passim*

rective gives computer programs an unusually "thick" scope of protection as to the underlying structure of programs.[147] However, the Directive defines interfaces necessary for interoperability as unprotectable ideas and principles,[148] even though they may be very important and commercially significant elements of program structure.

The sui generis decompilation provisions of the Directive reinforce the thick protection for most internal program structure because it is illegal under the Directive to decompile a program unless the decompiler is trying to get access to interface information.[149] 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.[150] The Directive

(1994). Within the framework set forth in the Manifesto, program interfaces would be industrial compilations of applied know-how eligible for a short period of exclusivity, following which others could use the interfaces, subject to an obligation to compensate the interface's developer. *See id.* at 2326–30, 2370. Treating interfaces as unprotectable ideas or principles avoids a problem that the Manifesto did not address, namely, the likelihood that firms would revise their interfaces to ensure continued exclusivity and thereby thwart compatibility with unlicensed parties' products.

[147] *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 EC Copyright Protection for Computer Programs: Are They More Different than They Seem?*, 13 J.L. & COM. 279, 292–97 (1994).

[148] *See* Software Directive, *supra* note 5, at 43–44.

[149] *See id.* at 45; van Rooijen, *supra* note 144, at 130–32 (discussing the balancing mechanisms embedded in the sui generis provisions of the Directive). The Directive's highly restrictive rules about decompilation are in stark contrast to the U.S. fair-use balancing approach. *See* Samuelson, *supra* note 147, at 285–92.

[150] 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* Case T-201/04, Microsoft Corp. v. Comm'n, 2007 E.C.R. II-3601, ¶ 276. 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

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.[151] This suggests that the Directive's intent is not to encourage reverse engineering activities, but rather to induce firms to license interface information on reasonable terms because if they don't, would-be interoperators will be able to lawfully reverse engineer the code to extract the information.[152]

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. 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.[153] 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.[154]

If, as this Article suggests, the Software Directive created 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.[155] European patent law, like U.S. patent law, does not allow ideas or principles to be patented.[156] Cha-

Commission thought trade secrecy would not be necessary for program details.

[151] *See* Software Directive, *supra* note 5, at 45. The Software Directive makes clear that it is not lawful to reverse engineer any parts of the program other than those that contain interface information. *Cf. id.*

[152] 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.

[153] *See* Software Directive, *supra* note 5, at 45.

[154] *Id.*

[155] The European Patent Office does not, however, interpret the Software Directive as precluding patents on program interfaces, for it has issued some patents on interface designs to Microsoft. *See* MICROSOFT PATENT MAPPING, *supra* note 133.

[156] *See* BAKELS & HUGENHOLTZ, *supra* note 77, at 28.

racterizing interfaces necessary to interoperability as ideas and principles would logically mean that the interfaces should be regarded as unpatentable under the Directive.[157] The Commission may have had this interpretation in mind when it denied that Microsoft owned patents on its interfaces.[158]

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 the appeal that Microsoft did have some IPRs in its interfaces.[159] The issue of whether the Software Directive excludes interfaces from patent protection has thus been left for another day.

Program interfaces are far more likely to be regarded as patentable subject matter under U.S. law. Many interface patents have been sought and issued under the very broad conception of patentable subject matter articulated in *State Street Bank & Trust Co. v. Signature Financial Group, Inc.*[160] That

[157] *See* Software Directive, *supra* note 5, at 45. Other functional design elements of programs, apart from interfaces, may be eligible for patents. There is less need, however, for firms to patent program structure in the EU, since the Directive provides such a thick layer of protection for program internals. Developers can create such internals without concomitant disclosure requirements. *Cf.* BAKELS & HUGENHOLTZ, *supra* note 77, at 22; *see also* Pamela Samuelson, Economic and Constitutional Influences on Copyright Law in the United States 4–10 (Oct. 20, 2000) (SSRN Working Paper Series), *available at* http://papers.ssrn.com/sol3/papers.cfm?abstract_id=234738 (explaining the differences between European and American protection of software).

[158] Case T-201/04, Microsoft Corp. v. Comm'n, 2007 E.C.R. II-3601, ¶ 278. Reinforcing this interpretation of the Software Directive is a provision in a directive proposed by the European Commission 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 [the Software] Directive . . . on the legal protection of computer programs by copyright, in particular under the provisions thereof in respect of decompilation and interoperability." Robert Bray, *The European Union "Software Patents" Directive: What Is It? Why Is It? Where Are We Now?*, 2005 DUKE L. & TECH. REV. 11, ¶ 22 (2005) (emphasis omitted). Thus, if it was lawful under the 1991 directive to reverse engineer a program to access interface information 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 interfaces are ideas and principles. Although the European Parliament ultimately rejected the proposed software patent directive, see *id.* ¶ 27, it remains to be seen whether the courts will interpret the Software Directive as precluding patent as well as copyright protection for program interfaces.

[159] Case T-201/04, Microsoft Corp. v. Comm'n, 2007 E.C.R. II-3601, ¶ 283.

[160] 149 F.3d 1368 (Fed. Cir. 1998); *see also* AT&T Corp. v. Excel Commc'ns, Inc., 172 F.3d 1352, 1356–58 (Fed. Cir. 1999) (discussing *State Street*).

decision considered everything under the sun made by humans to be patentable subject matter as long as it produces a "useful, concrete, and tangible result."[161] Program interfaces, as human-made designs that result in information being exchanged across ICT system boundaries, seemingly produced a useful, concrete, and tangible result.

The Federal Circuit has, however, recently repudiated the *State Street* conception of patentable subject matter. This occurred after several members of the U.S. Supreme Court questioned the Federal Circuit's overbroad view of patent subject matter,[162] which was plainly inconsistent with prior Supreme Court rulings.[163] Seemingly emboldened by the higher court's questions, the U.S. Patent & Trademark Office (PTO) resumed its earlier practice of rejecting applications for claiming unpatentable subject matter.[164] The Federal Circuit has recently affirmed PTO rejections of three patent claims on subject matter grounds,[165] including one en banc ruling that a method for

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

[162] The Supreme Court granted certiorari in *Lab. Corp. v. Metabolite Labs.*, 543 U.S. 1185 (2005), to review the patentability of a method of measuring the amount of a certain chemical in a patient's bloodstream for the purpose of diagnosing the patient with a vitamin deficiency. The Court ultimately decided that the writ had been improvidently granted, apparently because the subject matter issue had not been presented cleanly below. *Lab. Corp. v. Metabolite Labs.*, 548 U.S. 926 (2005). However, Justice Breyer wrote a powerful dissent, joined by two other Justices, that questioned the Federal Circuit's patentable-subject-matter standard. *Lab. Corp. v. Metabolite Labs.*, 548 U.S. 124, 136–37 (2005) (Breyer, J., dissenting). He concluded that a literal reading of *State Street*'s standard would lead to the granting of patents for processes that the Court has found unpatentable. *Id.* 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 Section 271(f)*, 50 COMM. ACM, June 2007, at 15, 18–19.

[163] *See* Brief for Consumers Union, Electronic Frontier Foundation, & Public Knowledge for Hearing *En Banc* as Amicus Curiae at 22–24, *In re* Bilski, 545 F.3d 943 (2008) (No. 07-1130), *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).

[164] Since 2006, the PTO has rejected numerous applications for failure to claim patentable subject matter. *See, e.g.*, *Ex parte* Yang-Huffman, No. 2007-2130, 2007 WL 2899992, at *2 (BPAI 2007) (rejecting a claim for a method for dynamic configuration of information); *Ex parte* Gosby & Ito, No. 2007-3941, 2007 WL 2843739, at *2 (BPAI 2007) (rejecting a claim for a method of document analysis and retrieval); *Ex parte* Gutta, 84 U.S.P.Q.2d 1536, 1537 (BPAI 2007) (rejecting a claim for a method for evaluating closeness of two items).

[165] *In re* Comiskey, 499 F.3d 1365, 1368 (Fed. Cir. 2007) (concluding that a method for conducting arbitrations through the use of legal documents is an unpatentable process); *In re* A.C.M. Nuijten, 500 F.3d 1346, 1348 (Fed. Cir. 2007) (holding that an encoded signal is unpatentable subject matter).

hedging the risk of price fluctuations for commodities claimed unpatentable subject matter.[166] The Federal Circuit decided that for a claim to satisfy the subject matter requirement, it must either be "tied to a particular machine or apparatus" or transform an article from one physical state to another.[167]

It is too early to know whether these changes in subject matter standards will make interface patents vulnerable to patentability challenges, but this is possible insofar as they are for methods of representing data, methods of calculating numbers, or methods of information exchange.[168] The Federal Circuit is, however, very unlikely to rule that software innovations are per se unpatentable, as they can be tied to machines and have transformative effects.[169] 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 for excluding interfaces essential to interoperability from patent protection.[170]

Congress could, of course, legislate an exclusion of interfaces from patent protection.[171] But at this point, there is insufficient momentum or consensus in the U.S. policy arena about the importance of interoperability or patents as an impediment to interoperability. It is thus likely that Congress would consider excluding patents for interface innovations in order to facilitate interoperability.[172]

[166] *In re* Bilski, 545 F.3d 943, 949 (Fed. Cir. 2008) (en banc) (abrogating *State Street*). That the Federal Circuit is not yet of one mind about standards for patentable subject matter is evident from the five opinions in that case. However, it is notable that all but one of the twelve judges believed that Bilski's method was unpatentable. *Id.* at 976–98 (Newman, J., dissenting).

[167] *Id.* at 954.

[168] *See* Samuelson, Benson *Revisited*, *supra* note 45, at 1032–62 (setting forth arguments against the patentability of such inventions).

[169] The Court might decide that program code is unpatentable subject matter and reaffirm the unpatentability of broad abstract algorithms, as in *Benson*. However, functional designs and processes embedded in software, including interface techniques, are likely to remain patentable subject matter.

[170] *See, e.g.*, Verizon Servs. Corp. v. Vonage Holdings Corp., 503 F.3d 1295, 1310 (affirming the injunction for infringement of a patent on an interconnection technique against the leading VoIP provider, and rejecting the argument that damages would adequately protect Verizon's interests).

[171] S*ee* Gerald J. Mossinghoff, *Remedies under Patents on Medical and Surgical Procedures*, 78 J. PAT. TRADEMARK OFF. SOC'Y 789 (1996) (discussing congressional proposals to exclude medical and surgical procedures from patent subject matter).

[172] In the free and open source software community, there is a strong con-

One commentator has, however, proposed to limit the exercise of interface patents through use of a fair use balancing test.[173] A first factor would consider the nature of the alleged infringer's advance of the art, under which makers of complementary products would be more favored than makers of competing platforms.[174] The purpose of the alleged infringer's use would be a second factor, under which reverse engineering to make a new product would be more favored than the marketing of a direct competing product.[175] A third factor would assess whether market failure had inhibited licensing, under which courts would pay attention to the patentee's reasons for refusing to license the interface patent or whether externalities had produced market failure.[176] The impact of a fair use ruling on incentives to invest in innovation and social welfare would be a fourth factor.[177] A fifth factor would focus on how much of an advance over the prior art the patented interface innovation represents.[178] A finding of fair use would in some cases result in the alleged infringer's being able to make free use of the invention, although sometimes the use might be "fared."[179]

cern that interface and other software patents are 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, *available at* http://searchenterpriselinux.techtarget.com/news/article/0,289142,sid39_gci1059267,00.html; Foundation for a Free Information Infrastructure, Software Patents in Action, http://eupat.ffii.org/patents/effects/index.en.html (last visited Apr. 28, 2009) (providing a compilation of news stories and case studies illustrating the impacts of software patents on the software industry, some giving examples of software patents that have impeded interoperability).

[173] *See* O'Rourke, *supra* note 96, at 1203–09.

[174] *See id.* at 1230.

[175] *See id.* at 1230–31.

[176] *Id.* at 1231 ("[A] refusal to license application developers is more suspect than the refusal to license the maker of a competing operating system. Failure to license application developers is both particularly troubling and likely to occur when the patentee also competes in the application market . . . .").

[177] *Id.* at 1233–34. Prof. O'Rourke argues that there will generally be high social benefit with little market harm when the infringer has developed a complementary product. Makers of directly competing products would not necessarily be excluded from being deemed fair users, but courts should assess the potency of network effects and whether the patentee has already reaped substantial rewards. *Id.*

[178] In particular, Prof. O'Rourke recommends that courts consider whether the patentee is exerting disproportionate leverage given the level of innovation in the interface. *Id.*

[179] *Id.* at 1234–35. That is, the fair user would have to pay a royalty to the patentee.

A third possible strategy for restricting interface patents is to allow them to issue, but deem their use noninfringing if essential for achieving interoperability. During the period when the European Parliament was considering whether to adopt a directive on patenting of software-related inventions, a proposal was made that use of any patents that "read" on interfaces should be deemed noninfringing insofar as there was no equally efficient or effective alternative nonpatented way to achieve interoperability.[180] The European Parliament did not adopt a software patent directive, so this provision was not adopted.[181]

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.[182] Section 287(c) of the U.S. Patent Act immunizes doctors from liability for using patented medical or surgical procedures to treat patients.[183] If a strong social consensus developed in favor of in-

[180] The proposed Article 6a 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 158, ¶ 22.

[181] Foundation for a Free Information Infrastructure, European Parliament Says No to Software Patents, http://wiki.ffii.org/Ep050706En (last visited Apr. 28, 2009).

[182] *See* H.R. 2365, 110th Cong. (2007) (providing that 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. (2007) (proposing immunity from patent infringement liability for firms supplying repair parts).

[183] 35 U.S.C. § 287(c) (2006). This provision was adopted after the American Medical Association and several other organizations of physicians lobbied for immunity after a surgeon was sued for infringing another surgeon's patented technique for cataract surgery. Mossinghoff, *supra* note 171, 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). *Id.* 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." WORLD TRADE ORGANIZATION, AGREEMENT ON TRADE-RELATED ASPECTS OF INTELLECTUAL PROPERTY RIGHTS 332 (1971), http://www.wto.org/english/docs_e/legal_e/27-trips.pdf. Because the overwhelming majority of patents in respect of medical procedures are owned by biotechnology and pharmaceutical firms, none of which sue doctors for treating patients, Mossinghoff argues that § 287(c) does not conflict with the normal exploitation of these patents. Mossinghoff, *supra* note 171, at 796–797. *But see* Emily C. Melvin, Note, *An Unacceptable Exception: The Ramifications of Physician Immunity from*

teroperability, Congress might well adopt a similar rule for immunizing use of patents as to interfaces essential to interoperability.[184]

Finally, one commentator has argued that owners of patents on interfaces should be deemed to have misused patents insofar as they use the patents as lock-out devices.[185] Nintendo's success in asserting its patented authentication method to stop Atari Games from making and selling games that could run on the Nintendo platform was, in her view, an unlawful extension of the patent's scope since, in essence, it created an unlawful tying arrangement between the Nintendo console and Nintendo-licensed games.[186] 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, even though the games were not within the scope of the patent.[187] 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. Adjusting Patent Policy Levers Affecting Interoperability and Other Patent Reforms

There are several ways that patent rules can be tailored to lessen risks that patents on interface techniques will impede interoperability. One example would be to heighten the standard of nonobviousness for ICT interface patent claims.[188] This proposal recognizes that firms may seek patents for interface designs for anticompetitive 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 plat-

Medical Procedure Patent Infringement Liability, 91 MINN. L. REV. 1088, 1089 (2007) (challenging both the wisdom of § 287(c) and its compatibility with TRIPs).

[184] It may be more difficult to justify an interoperability exception to enforcement of interface patents under TRIPs, however, because a normal exploitation of such patents may include licensing them.

[185] Cohen, *supra* note 116, at 1182–83.

[186] *Id.*

[187] 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. *Id.*

[188] *Id.* at 1152–81.

form).[189] To ensure that patents are being issued only to truly inventive interfaces, the PTO could apply an "innovative programmer" standard to judging patentability.[190] Under this standard, fewer interface patents would likely issue.

Although patents on ICT interfaces have not (yet) been subjected to a higher nonobviousness 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.[191] Other commentators have shown that many policy levers in patent law are available to respond to industry-specific considerations.[192] The PTO probably has inherent authority to scrutinize interface patents more carefully than others. If the PTO came to perceive interface patents as potentially being sought for anticompetitive reasons, that might well justify a closer look.

A second, though more indirect, way to tailor patent rules to facilitate interoperability would be to treat reverse engineering of an ICT interface for the purpose of obtaining access to interface information as noninfringing.[193] Some commentators have recommended allowing firms to reverse engineer program code to facilitate the development of interoperable programs, even if it was necessary to use a patented invention in the course of reverse engineering.[194] A number of patent doctrines

[189] *Id.* at 1152–53.

[190] *Id.* Cohen thinks that the 10NES patent would have been invalid under this heightened standard. *Id.* at 1153, 1162.

[191] *See, e.g.*, John R. Allison & 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). John Allison and Emerson Tiller mention that The Business Method Patent Improvement Act, H.R. 5364, 106th Cong. (2d Sess. 2000), initially called for heightening the nonobviousness standard for business method patents. *Id.*

[192] Burk & Lemley, *supra* note 75, at 1630.

[193] It is, of course, a separate question whether reimplementing the interface or other patented design in a follow-on product would infringe the patent. With traditional manufactured products, one can purchase an instance, take it apart, and study its internal design without worrying about patent infringement. One cannot reverse engineer computer software, however, without making patented components embodied in it. Prof. O'Rourke has argued that even reimplementation of a patented interface could be deemed fair use. *See* O'Rourke, *supra* note 96, at 1230–35.

[194] *See* Cohen et al., *supra* note 76, at 16–37. "Reverse engineering promotes the fundamental patent policies of disclosure and enablement, ensures that patents will not be leveraged to protect unprotectable components of

could serve as possible bases for this policy,[195] given that the social interest in promoting interoperability is no less important in patent law than in copyright and trade secrecy law.[196]

Another doctrinal development that would promote greater ICT interoperability would be judicial refusal to enforce antireverse engineering clauses of license agreements insofar as these agreements conflict with public policies favoring interoperability in intellectual property law.[197]

Requiring more meaningful disclosure about interface techniques in any patent for them would also promote interoperability. In theory, reverse engineering should be unnecessary to discern interface innovations, as the patent should reveal information necessary for someone skilled in the art to make them. However, it is well-known that patents for software innovations generally disclose relatively little.[198] Firms may claim the interface technique at high levels of abstraction so that an ordinary programmer would not be able to create interoperable components because the interface implementation details are kept as trade secrets.

To respond to this concern, patent examiners could be more rigorous about requiring meaningful disclosure as to interface techniques. Yet, for already issued interface patents, some reverse engineering may be necessary to extract interface information. Such reverse engineering to obtain information that should have been disclosed in the patent should not be deemed infringing.

software, preserves the balance sought by the intellectual property system as a whole, and also helps patentees to enforce their rights." *Id.* at 22; *see also* O'Rourke, *supra* note 96, at 1212.

[195] Cohen et al., *supra* note 76, 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).

[196] *Id.* at 27–28. Julie Cohen and Mark Lemley also recommend a narrow scope for the doctrine of equivalents in software-related patent cases, to address widespread and legitimate concerns about the low quality of patents in this field. *Id.* at 37–56.

[197] *See, e.g.*, Samuelson & Scotchmer, *supra* note 85, at 1626–30; *see also* 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, 555–58 (1992); Mark I. Koffsky, Note, *Patent Preemption of Computer Software Contracts Restricting Reverse Engineering: The Last Stand?*, 95 COLUM. L. REV. 1160, 1160 (1995). *But see* Bowers v. Baystate Techs., Inc., 320 F.3d 1317, 1326–27 (Fed. Cir. 2003) (affirming enforcement of an antireverse engineering clause in a software shrinkwrap license).

[198] *See, e.g.*, JAMES BESSEN ET AL., PATENT FAILURE, chs. 7, 9 (2008).

Certain patent reform measures could also ensure 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 nonobviousness standard for attaining patent protection,[199] and 2) a more cost-effective way to challenge invalid patents than the litigation and re-examination procedures under current patent law.[200]

These two reforms are interlinked because when the nonobviousness standard is too low, some patents will have issued that should not have; yet the cost of litigation is so high, that some invalid patents may not be challenged that should be. In *KSR Int'l, Co. v. Teleflex Inc.*,[201] the Supreme Court rejected the Federal Circuit's test for nonobviousness—which required proof of a "teaching, suggestion, and motivation" in the prior art—as insufficiently rigorous.[202] 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, indeed, been a key component of the patent reform legislation that has been pending before Congress in recent years.[203]

While reinvigorating the nonobviousness standard and an improved post-grant review process are reforms that are not specifically aimed at interface patents, 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 tri-

[199] FED. TRADE COMM'N, TO PROMOTE INNOVATION: THE PROPER BALANCE OF COMPETITION AND PATENT LAW AND POLICY 10–12 (2003) [hereinafter FTC Report].

[200] *Id.* at 7–8.

[201] 550 U.S. 398 (2007). A patent reform reinvigorating the nonobviousness standard would have been difficult to achieve in Congress because patent lawyers and various industry groups have profoundly different views on how rigorous the nonobviousness standard should be. The Supreme Court's *KSR* decision, though, 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, Feb. 2007, at 19.

[202] *KSR*, 550 U.S. at 419.

[203] H.R. 2365, 110th Cong. (2007).

vial 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 of the patented invention might also have implications for lowering the risk of substantial liability as to minimally innovative interface techniques.[204]

C. EMPLOYING LIABILITY RULES FOR USE OF INTERFACE PATENTS

Some commentators and policymakers have called for a liability rule approach to patents on interfaces.[205] This would allow unlicensed persons to implement patented interfaces to achieve interoperability as long as these persons offered reasonable compensation to the patentee. A liability rule approach can be implemented in a number of ways.

One commentator has proposed that courts should withhold injunctive relief for infringement of patents on interfaces essential for interoperability.[206] He draws upon the Supreme Court's ruling in *eBay, Inc. v. MercExchange, L.L.C.*[207] which rejected the Federal Circuit's rule that courts must virtually always issue injunctions in patent cases.[208] The Court in *eBay* observed that under traditional principles of equity, injunctions should not issue unless the plaintiff 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

[204] 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* Patent Reform Act of 2007, H.R. 1908, 110th Cong. § 5(b)(2) (2007). This proposal has, however, proven to be controversial. *See, e.g.*, Posting of Dennis Crouch to Patently-O, http://www.patentlyo.com/patent/2007/05/patent_reform_2_1.html (May 25, 2007, 07:21 EST).

[205] For a discussion of liability verses property rules in respect of information resources, see Mark A. Lemley & Philip J. Weiser, *Should Property or Liability Rules Govern Information?*, 85 TEX. L. REV. 783 (2007); J.H. Reichman, *Of Green Tulips and Legal Kudzu: Repackaging Rights in Subpatentable Innovation*, 53 VAND. L. REV. 1743 (2000). *See also* Guido Calabresi & A. Douglas Melamed, *Property Rules, Liability Rules, and Inalienability: One View of the Cathedral*, 85 HARV. L. REV. 1089, 1106–11 (1972).

[206] *See* Peter Lee, *The Evolution of Intellectual Infrastructure*, 83 WASH. L. REV. 39, 104 (2008).

[207] 547 U.S. 388 (2006).

[208] *Id.* at 393–94.

defendant, and 4) the public interest would not be disserved thereby.[209]

Justice Kennedy's concurrence in *eBay*, which was joined by three other Justices,[210] 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 patent."[211] This problem is especially acute "[w]hen the patented invention is but a small component of the product the companies seek to produce and the threat of an injunction is employed simply for undue leverage in negotiations . . ."[212] In such cases, "legal damages may well be sufficient to compensate for the infringement and an injunction may not serve the public interest."[213]

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.[214] 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.'"[215] 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 . . ."[216] But holders of patents on interoperability standards can potentially "hold an industry hostage by demanding crippling royalties."[217] Infringers of patents on interoperability standards should be eligible for compensation under *eBay*, Nokia argued, but not injunctive relief.

[209] *Id.* at 391.

[210] *Id.* at 395. Justices Breyer, Souter, and Stevens joined in the opinion. *Id.*

[211] *Id.* at 396.

[212] *Id.*

[213] *Id.* at 396–97. The President of the EU has reportedly also 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 1 (2008).

[214] *See* Brief of Amicus Curiae Nokia Corp. in Support of Petitioners at 4, eBay, Inc. v. MercExchange, L.L.C., 547 U.S. 388 (2006) (No. 05-130), 2006 WL 235005.

[215] *Id.*

[216] *Id.* at 12.

[217] *Id.*

In accord with this argument, one commentator 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."[218] Patents on interfaces essential for interoperability are among the infrastructural inventions that should meet this test.[219] This proposal aims 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"[220] 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.

Japanese policymakers have taken a different approach to establishing a liability rule for patents essential to interoperability. In 2005, the Japanese Ministry of Economy, Trade, and Industry (METI) established a "Study Group on the Legal Protection of Software and Promotion of Innovation," which expressed serious concerns about the exclusionary potency of interface patents.[221] 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 a lock-in effect on users, the granting of patents may create unduly powerful exclusive rights."[222] Even though the Study Group recognized that most patents are exercised in a manner that promotes innovation, it observed that interface patents posed risks of adverse effects on innovation.[223] The Study Group encouraged the use of Creative Commons-type licensing for patents affecting

[218] *See* Lee, *supra* note 206, at 46; *see also* Frischmann, *supra* note 140, at 956.

[219] *See* Lee, *supra* note 206, at 46.

[220] *Id.* at 109. There are two other ways that a liability rule could be implemented as to interface patents. Prof. O'Rourke has proposed development of a fair use defense in patent law that might result in a payment of royalties to the interface patent holder. *See* O'Rourke, *supra* note 96, at 1233–34. The U.S. government could also 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. § 1498 (2006) or under its eminent domain powers.

[221] *See* ISID INTERIM REPORT, *supra* note 4.

[222] *Id.*

[223] *Id.*

interoperability, but also recommended compulsory licensing and enhanced application of antimonopoly law as policy responses to such patents.[224]

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 rights.[225] "[W]here 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 its adverse competitive effect[.]"[226] Whether a particular refusal to license an interface patent is an abuse of IP rights will, however, be determined through a comprehensive assessment of the facts on a case by case basis, taking many factors into account.[227]

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.[228] 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 nondiscriminatory (RAND) terms.[229]

[224] *Id.*

[225] JAPANESE MINISTRY OF ECON., TRADE, AND INDUS., INTERPRETATIVE GUIDELINES FOR ELECTRONIC COMMERCE AND INFORMATION PROPERTY TRADING 192–193 (2007) (Tentative Translation), *available at* http://www.meti.go.jp/english/information/data/IT-policy/interpretative_guidelines_on_ec070628.pdf.

[226] *Id.* at 193 n.36.

[227] *Id.* at 196. METI offered the following examples of potential harm from interface patents: patents that implicate interoperability of software supporting "critical infrastructure"; "universal software that is widely used in our society"; information services "in which particular individuals participate," such as online auctions; and 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. *Id.* at 201 n.51.

[228] *See* Bray, *supra* note 158, ¶ 16.

[229] FOUNDATION FOR A FREE INFORMATION INFRASTRUCTURE, PLENARY AMENDMENTS TO THE PROPOSED SOFTWARE DIRECTIVES 2–3 (2005). This proposal was not included in the Council's March 2005 common position. *See* Bray, *supra* note 158, ¶¶ 27–28.

A fourth liability-rule initiative that would affect interface patents, although it is not directly aimed at them, is a proposed European "Soft IP" right.[230] 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.[231] Lawyers for IBM hope that this regime will be adopted and widely used by firms seeking patents on interfaces and other software-related inventions.[232]

D. Invoking Competition and Antitrust Law to Facilitate Interoperability

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.[233] The first arose when the Commission initiated a case against IBM Corp. in the 1980s, and the second when the Commission took action against Microsoft in the early years of the twenty-first century.[234] The latter case resulted in the Commission ordering Microsoft to provide information necessary to interoperability to developers of competing technologies.[235] Microsoft was also forced to disclose information about its interfaces and license IP rights in them as a remedy for violating U.S. antitrust laws in the 1990s.[236]

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

[230] *See, e.g.*, James Nurton, *IBM Flies Soft-IP Community Patent Plan*, MANAGING INTELLECTUAL PROPERTY, July 31, 2007, http://www.managingip.com/Article.aspx?ArticleID=1398998.

[231] *Id.*

[232] Telephone Interview with David Kappos, Vice President & Assistant Gen. Counsel for Intellectual Prop., IBM Corp. (Apr. 23, 2008).

[233] *Cf.* F.M. Scherer, *Microsoft and IBM in Europe*, 84 ANTITRUST & TRADE REG. REP. 65, 65–66 (Jan. 23, 2003) (providing an overview of the two cases).

[234] *See id.*

[235] Commission Decision, *supra* note 6, at 299–300.

[236] New York v. Microsoft Corp., 224 F. Supp. 2d 76 app. at 269 (D.D.C. 2002).

developed.[237] The IBM case also affected the Commission's perception of the competitive harms likely to flow from Microsoft's withholding of interface information from Sun and others and of the likely competitive benefits of forcing Microsoft to disclose information about its communications protocols to its competitors.[238]

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.[239] 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.[240] Even after IBM started unbundling software and peripherals, under pressure from antitrust authorities,[241] it did not publish its interfaces, but rather licensed them as trade secrets on royalty-bearing terms.[242] Although licensing interface information did facilitate the development of IBM-compatible technologies, IBM upset many of its licensees by making frequent changes to its interfaces, which caused the licensees' previously compatible technologies to be less compatible or wholly incompatible.[243] European competition law authorities

[237] 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 Commission's Competition Directorate. *See* Software Directive, *supra* note 5, at 44–45; *cf.* Alan K. Palmer & Thomas C. Vinje, *The EC Directive on the Legal Protection of Computer Software: New Law Governing Software Development*, 2 DUKE J. COMP. & INT'L L. 65, 71–78 (1992) (providing the procedural history of the Software Directive). The initial draft of the Directive did not contain these provisions. *Cf. id.* at 67–71 (discussing the initial drafts of the Software Directive).

[238] *Cf.* Commission Decision, *supra* note 6, at 198–200 (referring to a previous case against IBM in justifying the order in the Microsoft case); Scherer, *supra* note 233, at 65–66 (discussing similarities between the Commission's case against Microsoft and its earlier case against IBM over delayed disclosures of and changes to interfaces).

[239] *See* Maria Lillà Montagnani, *Predatory and Exclusionary Innovation: Which Legal Standard for Software Integration in the Context of the Competition versus Intellectual Property Rights Clash?*, 37 INT'L REV. INTELL. PROP. & COMPETITION L. 304, 310 n.17 (2006) ("[I]n the earlier IBM cases, more relevance was given to trade secrets law, since IBM, after having integrated accessories, used to refuse to disclose the new interfaces by claiming them as secret.").

[240] *Id.* at 312.

[241] *See* BAND & KATOH, *supra* note 18, at 22 n.30.

[242] *Id.*

[243] *See* Montagnani, *supra* note 239, at 310–11 (discussing allegations

charged IBM with abusing its dominant position by, among other things, changing interfaces in a manner that rendered IBM-compatible peripherals inoperable.[244] IBM settled the lawsuit by agreeing to pre-disclose changes to its interfaces to aid other firms in adapting their products in a timely manner.[245]

A decade or so after the IBM case, the European Court of Justice handed down the *Magill* and *IMS Health* decisions that established that a dominant firm's refusal to license IP rights can, in exceptional circumstances, constitute an abuse of dominant position under EU competition law.[246] *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 all 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."[247]

The Commission applied and adapted this test in its competition law proceeding against Microsoft in the early 2000s.[248]

that IBM had engaged in predatory innovation); *see also In re* IBM Peripheral EDP Devices Antitrust Litig., 481 F. Supp. 965, 1003–04 (N.D. Cal. 1979) (peripheral makers charged that IBM's changes to interfaces were anticompetitive because they thwarted compatible products). Industry groups made similar claims against Microsoft to the EC. *See* Case T-201/04, Microsoft Corp. v. Comm'n, 2007 E.C.R. II-3601, ¶ 282.

[244] *See* Montagnani, *supra* note 239, at 310–11.

[245] IBM also tried to block firms, notably Fujitsu, from developing platforms capable of interoperating with applications written for IBM computers. *See* BAND & KATOH, *supra* note 18, 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 44, at 243–47. 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. *Cf.* Case T-201/04, Microsoft Corp. v Comm'n, 2007 E.C.R. II-3601 ¶ 709–11 ("IBM's commitment to the Commission in 1984 was not substantially different from what Microsoft was ordered to do.").

[246] Case C-418/01, IMS Health v. NDC Health, 2004 E.C.R. I-5039, I-5087–88; Joined Cases C-241 & 242/91, Radio Telefis Eireann v. Comm'n, 1995 E.C.R. I-743, I-759 (the "Magill" case).

[247] Case T-201/04, Microsoft Corp. v Comm'n, 2007 E.C.R. II-3601 ¶ 116; *see also* François Lévêque, *Innovation, Leveraging and Essential Facilities: Interoperability Licensing in the EU Microsoft Case*, *in* ANTITRUST, PATENTS AND COPYRIGHT: EU AND US PERSPECTIVES 103, 104 (François Lévêque & Howard Shelanski eds., 2005) (providing the *Magill/IMS Health* test).

[248] The Commission adapted the *Magill/IMS* test in that it focused on the refusal to license as an impediment to the development of new features

Microsoft was charged with abuse of dominant position because it was unwilling to supply enough information and supporting technologies to enable Sun Microsystems to adapt its Solaris WGS-OS so that it could be fully compatible with Microsoft's Windows-based OS technologies, especially as to the Active Directory technologies that include protocols for synchronizing operations among various domains, including efficient exchanges of update information, in distributed networked environments.[249] 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 thereafter withheld similar information in order to gain additional market share at the expense of its rivals, thereby abusing its dominant position.[250]

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 Sun's ability to remain a viable competitor in the WGS-OS market.[251] The Commission believed that Microsoft's refusal

rather than to the emergence of a new product. *See* Lévêque*, supra* note 247, at 104–08 (discussing this adaptation).

[249] *See* Case T-201/04, Microsoft Corp. v Comm'n, 2007 E.C.R. II-3601 ¶¶ 2–20. Although the Commission relied heavily on *Magill* and *IMS Health*, the Commission's charges were not based on Microsoft's unwillingness to license its IPRs in the interfaces, but rather its unwillingness to supply detailed interface information to Sun and others. *Cf. id.* ¶ 2 (describing Sun Microsystems' request for "reference implementation and such other information as is necessary to insure, without reverse engineering, that COM objects and the complete set of Active Directory technologies will run in full compatible fashion on Solaris"). For a discussion of the Active Directory technologies and the amount of disclosure required to implement them, see Page & Childers, *supra* note 91, at 341–43.

[250] *See* Commission Decision, *supra* note 6, at 157–58. Sun and other makers of UNIX-based WGS-OS systems had initially obtained access to Windows interface information through a license Microsoft granted to AT&T to facilitate UNIX compatibility with Windows technologies at a time when Microsoft was not in the WGS-OS market. *See* Case T-201/04, Microsoft Corp. v Comm'n, 2007 E.C.R. II-3601 ¶¶ 569–73. Once Microsoft entered that market, it no longer had the same incentive to supply detailed interface information to AT&T, and that license was not renewed. *See id.* at 573. As Page & Childers note, Microsoft had never disclosed to AT&T or any other firm the then-newly developed Active Directory technologies, access to which Sun sought in 1998. Page & Childers, *supra* note 91, at 338–39. Information supplied under the AT&T license facilitated emulation of the Windows NT server software, an earlier generation technology. *Id.* at 339.

[251] Commission Decision, *supra* note 6, at 146, 294. The Commission also ruled against Microsoft on a separate charge as to abuse of dominant position in respect to its media player software. *Id.* at 294–95.

to supply this information threatened to eliminate competition in the WGS-OS market because of powerful network effects that were tipping this market to Microsoft's product.[252] It cited evidence that customers preferred many features of other WGS-OS systems; yet, Microsoft had an advantage over Sun because customers also valued compatibility with Windows technologies.[253] 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.[254] 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.[255]

To remedy this abuse, the Commission ordered Microsoft to prepare sufficiently detailed specifications of its communications protocols 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 nondiscriminatory (RAND) terms, and to update the information promptly as its protocols changed.[256] Microsoft was also ordered to establish an evaluation mechanism to ensure compliance with the Commission's order.[257]

Microsoft appealed the Commission's order to the European Court of First Instance (CFI), arguing, among other things, that the Commission had misinterpreted the interope-

[252] *Id.* ¶ 980. Prof. Lévêque reasons that once Microsoft had attained a certain market share in the workgroup 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. Lévêque*, supra* note 247, at 113–14. This, in turn, would cause customers and ISVs to be concerned about being stranded. *Id.* The market would then tip to Microsoft, with network effects to finish the work of killing off the competition. *Id.*

[253] Case T-201/04, Microsoft Corp. v Comm'n, 2007 E.C.R. II-3601 ¶¶ 405–09.

[254] *See id*. ¶¶ 631, 647.

[255] *Id.* ¶¶ 697–98. Professor Lévêque has questioned whether it was appropriate to make Microsoft bear the burden of proof on this issue. Lévêque*, supra* note 247, at 121. I agree.

[256] Case T-201/04, Microsoft Corp. v Comm'n, 2007 E.C.R. II-3601 ¶ 48.

[257] *Id.*

rability provisions of the Software Directive,[258] that its ownership of IPRs, especially key patents, in the interfaces provided an objective justification for its refusal to supply extensive interoperability information to its competitors,[259] and that unless the company had freedom to choose how to exercise its IPRs in its protocols, it would have inadequate incentives to invest in research and development to improve its products.[260]

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.[261] 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).[262] 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 designed for the platform whose interface information was at issue.[263]

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.[264] It objected to being required to give competitors extensive in-

[258] *Id.* ¶ 121.

[259] *Id.* ¶¶ 115–24. Microsoft claimed copyrights in interface documentation, trade secret protection for the interfaces themselves, and patents on some communications protocols. *See id.* ¶¶ 112, 115, 267–73.

[260] *Id.* ¶ 274.

[261] *Id.* ¶¶ 108, 225–26.

[262] *Id.* ¶ 108.

[263] Robert Hart characterized one-way interoperability as enabling multi-vendor compatibility and two-way as enabling plug-replaceability. *Cf.* Robert J. Hart, *Interoperability Information and the Microsoft Decision*, 2006 EUR. INTELL. PROP. REV. 361, 361 ("[T]he Commission's decision . . . aims to achieve plug replaceability—a standard of interoperability to which the EC Software Directive does not aspire. Rather, the author's reading of the Directive is that the goal of the interoperability . . . is a more modest one, namely multi-vendor interoperability."). Hart asserted that the Directive was only intended to support multi vendor compatibility. *Id.* The text of the Directive, however, does not support this theory. *See* Software Directive, *supra* note 5, at 45.

[264] *See* Case T-201/04, Microsoft Corp. v Comm'n, 2007 E.C.R. II-3601 ¶¶ 118–23.

formation that would allow them to "clone" its technologies.[265] Microsoft argued that this forced disclosure would be harmful to investments in innovation in the WGS-OS market.[266] 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.[267] 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.[268]

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.[269] The Directive extends protection to the latter, but not to the former.[270] The Commission pointed out that "the word '*inter*-operability,' by its very nature, relates to a two-way relationship," and that Microsoft's definition "is difficult to reconcile with the wording of the definition of interoperability [in the Software Directive]: '*the ability to exchange information and mutually to use the information which has been exchanged*.'"[271] The Commission's order did not require Microsoft to disclose source code, algorithms, or other internal design details of Microsoft's technologies, but only program interfaces.[272] For this reason, the Commission believed that competitors would not be able to clone Microsoft's technologies, but only to interoperate with software developed for Windows. In the Commission's view, the order merely required Microsoft to comply with the legislatively endorsed policy favoring interoperability embedded in the Software Directive.[273]

[265] *Id.* ¶ 110; *see also* Page & Childers, *supra* note 91, at 333, 351–52 (asserting that the information disclosed to Samba would facilitate cloning and explaining the basis for Microsoft's claim that it would have to disclose certain algorithms and other program internals to aid others in achieving interoperability with its Active Directory technologies).

[266] *See* Case T-201/04, Microsoft Corp. v Comm'n, 2007 E.C.R. II-3601 ¶ 111.

[267] *See id.* ¶ 668.

[268] *See id.* ¶ 670.

[269] *Id.* ¶¶ 195–99.

[270] *See id.* ¶¶ 225–26.

[271] Commission Decision, *supra* note 6, ¶ 758.

[272] Case T-201/04, Microsoft Corp. v Comm'n, 2007 E.C.R. II-3601 ¶¶ 194–206.

[273] *Id.* ¶ 710. The Commission took similar action against IBM. *See* Scherer, *supra* note 233, at 65–66. That proceeding was eventually settled by

Before the CFI, the Commission denied that Microsoft had any IP rights in its interfaces.[274] It 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.[275] It questioned whether Microsoft's protocols were innovative enough to qualify for patent protection, although it also spoke of interfaces as ideas and principles.[276] And the Commission discounted Microsoft's trade secrecy claims because the value in this information lay in their secrecy, not in any innovation they might embody.[277] Yet, 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.[278]

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.[279] Both Microsoft and its competitors would, in its view, have ample incentives to invest in further improvements in their technologies in order to respond to and fuel consumer demand.[280] Because Sun and others had

IBM's agreement to disclose interface information to other firms in advance of releasing new systems into the market. *See id.* This experience informed the Commission's proposal for the interoperability provisions of the Software Directive. *See* van Rooijen, *supra* note 144, at 130; Palmer & Vinje, *supra* note 237, at 72 n.34.

[274] Case T-201/04, Microsoft Corp. v Comm'n, 2007 E.C.R. II-3601 ¶ 277.

[275] *Id.* ¶ 279.

[276] *Id.* ¶¶ 276–78. It is difficult to accept the Commission's view that there was no innovation in Microsoft's Active Directory technologies, which were at the core of the Commission's investigation, given the substantial complexity of designing protocols for synchronizing updates among domains in distributed networked environments and the inability of Sun and others to figure out how to implement these interfaces without information supplied by Microsoft. *Cf.* Page & Childers, *supra* note 91, at 335 (characterizing Active Directory as "one of Microsoft's most distinctive and innovative server technologies").

[277] *See* Case T-201/04, Microsoft Corp. v Comm'n, 2007 E.C.R. II-3601 ¶ 280.

[278] *Id.* ¶¶ 689–90.

[279] *See* Lévêque, *supra* note 247, at 117–18 (discussing the challenge of setting a "reasonable royalty" for this interface information, especially given the nondiscrimination requirement).

[280] *Cf.* Case T-201/04, Microsoft Corp. v Comm'n, 2007 E.C.R. II-3601 ¶ 654 (providing examples of innovations that benefited both the product developer and Microsoft). *But see* Page & Childers, *supra* note 91, at 354 (concluding that disclosure of interface information to Samba would harm Microsoft's incentives to invest in innovation).

invested in innovative WGS-OS designs during the period when Microsoft was supplying higher levels of interoperability information, the Commission believed that requiring Microsoft to disclose such information would not significantly dampen its or its competitors investments in innovation in the future.[281]

In September 2007, the CFI affirmed the Commission's order, holding that 1) the Commission's interpretation of the Software Directive was sound;[282] 2) under *Magill* and *IMS Health*, ownership of IPRs was not, by itself, an objective justification for refusal to license them to others;[283] 3) the exceptional circumstances required by *Magill* and *IMS Health* had been satisfied;[284] and 4) Microsoft had failed to prove that its incentives to invest in innovation would be diminished by the order, saying that the firm had provided only "vague, general, and theoretical arguments" in support of this claim.[285] The CFI pointed out that it was standard industry practice to license interface information,[286] and that Microsoft itself had agreed to provide interface information in settling litigations against it in the United States.[287] The Commission's order was, moreover, consistent with the Software Directive and the IBM settlement in a similar competition case in the mid-1980s.[288] Microsoft decided not to appeal the CFI ruling.[289]

Since then, Microsoft has disclosed considerable amounts of interface information under the Commission's order.[290] How-

[281] *See* Case T-201/04, Microsoft Corp. v Comm'n, 2007 E.C.R. II-3601 ¶ 658; *see also* Ashwin van Rooijen, *The Role of Investments in Refusals to Deal*, 31 WORLD COMPETITION: L. & ECON. REV. 63, 69–70 (2008) (discussing the investment incentives analysis in the CFI Decision in *Microsoft*).

[282] Case T-201/04, Microsoft Corp. v Comm'n, 2007 E.C.R. II-3601 ¶¶ 225–26.

[283] *Id.* ¶ 678.

[284] *Id.* ¶¶ 690–91.

[285] *Id.* ¶¶ 697–701.

[286] *Id.* ¶¶ 702, 710. Neither the Commission nor the CFI clarified whether it was a common industry practice for developers to disclose interface information to makers of functionally equivalent products, or only to developers of complementary products. The Commission should have provided some empirical data to support the implicit claim that both kinds of licenses were common.

[287] *Id.* ¶ 703.

[288] *Id.* ¶ 710.

[289] *See* Press Release, Microsoft Corp., Microsoft Statement on Compliance with European Commission 2004 Decision (Oct. 22, 2007), http://www.microsoft.com/Presspass/press/2007/oct07/10-22MSStatement.mspx.

[290] *See* Stephen Castle, *Microsoft Gets Record Fine and a Rebuke from Europe*, N.Y. TIMES, Feb. 28, 2008, at C3 (reporting that Microsoft had published 30,000 pages of previously secret source code for the Windows operating

ever, the Commission fined Microsoft $1.35 billion for failing to provide sufficiently detailed interoperability information.[291] Microsoft and the Commission have also 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 might affect the price which Microsoft can charge licensees for providing interoperability information.[292] Key differences between U.S. and EU antitrust/competition law cast doubt on whether U.S. antitrust authorities would pursue or U.S. courts would uphold similar claims against Microsoft. One of the 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 nondiscriminatory terms as long as doing so would not cause undue congestion or the like in providing access to that facility.[293]

The viability of the "essential facility" doctrine as a matter of U.S. antitrust law is uncertain after the Supreme Court decision in *Verizon Communications Inc. v. Law Offices of Curtis V. Trinko, LLP*.[294] Even assuming that *Trinko* did not deliver a death blow to that doctrine,[295] 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 interfaces and that unless it is able to recoup its R&D expenses, it will have too little incentive to invest in innovation.[296] U.S. courts have thus far been unwilling to hold that a

system to comply with the Commission's order, yet the Commission was unimpressed due to multiple unfulfilled promises to produce the code earlier).

[291] *Id.*

[292] Scott M. Fulton, III, *EU Threatens to Fine Microsoft for Lack of Innovation*, BETANEWS, Mar. 1, 2007, http://www.betanews.com/article/EU_Threatens_to_Fine_Microsoft_for_Lack_of_Innovation/1172764299.

[293] Lévêque, *supra* note 247, at 103–10, 120–21 (discussing the Commission's essential facility theory).

[294] 540 U.S. 398, 398–99 (2004) (holding that Verizon's unwillingness to provide certain services to local exchange competitors did not violate the antitrust laws). The Court took into account that Congress had already provided rules for access and the Federal Communications Commission and state regulators engage in close regulatory oversight of such matters. *Id.* at 411–13.

[295] *See, e.g.*, Michael A. Carrier, *Refusals to License Intellectual Property After* Trinko, 55 DEPAUL L. REV. 1191, 1207–08 (2006); *see also* van Rooijen, *supra* note 281, 63–77 (comparing U.S. and EU case law on essential facilities and refusals to deal).

[296] *See, e.g.*, *Trinko*, 540 U.S. at 407–08.

refusal to license IPRs to competitors is a violation of the U.S. antitrust laws.[297]

There is, however, some similarity between the Commission's order in *Microsoft* and the Supreme Court's ruling in *Aspen Skiing Co. v. Aspen Highlands Skiing Corp*.[298] In both cases, there was a history of sharing 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.[299] 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.[300] 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 United States, courts have sometimes ordered antitrust violators to license IPRs and/or disclose nonpublic information, such as interface specifications, to competitors.[301] The consent decree settling the U.S. antitrust case against Microsoft in the 1990s, for example, required Microsoft to disclose interface information and license pertinent IPRs to firms that might want to use them, even though U.S. antitrust authorities had not charged Microsoft with having

[297] *See, e.g.*, Herbert Hovenkamp et al., *Unilateral Refusals to License in the US*, *in* ANTITRUST, PATENTS AND COPYRIGHTS, *supra* note 247, at 12, 27–30.

[298] 472 U.S. 585 (1985).

[299] *Id.* at 609–11.

[300] *Trinko*, 540 U.S. at 407.

[301] *See, e.g.*, William E. Kovacic, *Designing Antitrust Remedies for Dominant Firm Misconduct*, 31 CONN. L. REV. 1285, 1304–05 (1999) (giving examples of licenses induced by antitrust oversight). This article was published as part of a symposium issue of the Connecticut Law Review on the remedy challenges posed by *United States v. Microsoft Corp*. *See* Robert M. Langer, *Symposium Introduction*: United States v. Microsoft, 31 CONN. L. REV. 1245, 1247 (1999); *see also* SOFTWARE & INFO. INDUS. ASS'N, ADDRESSING THE MICROSOFT CHALLENGE—RESTORING COMPETITION TO THE SOFTWARE INDUSTRY (1999), http://www.manishin.com/pressdocs/siia.pdf (providing a comprehensive analysis of remedy alternatives).

misused any patents on its interfaces or refused to license IPRs in its interfaces to competitors.[302] U.S. antitrust officials persuaded the judge overseeing the settlement to take a "forward-looking" approach to thwart possible Microsoft efforts to maintain the firm's monopoly in the Windows OS market by restricting access to interface information.[303]

One option that would have given other firms more access to Windows' interfaces would have been to require Microsoft to license its source code to those who wanted to make interoperable programs.[304] Although this would have promoted greater interoperability, some expressed concern this remedy would be more generous to competitors than was warranted by the antitrust violations.[305] A second option was to require Microsoft to disclose and license interface information to firms wanting to develop interoperable technologies. Microsoft agreed to the latter in settling the U.S. antitrust case.[306]

U.S. antitrust authorities and the European Commission have been frustrated in their efforts to promote competition through forced disclosure of interface information because compulsory licensing of IPRs and/or knowhow, such as interface information, is challenging as an antitrust remedy. It requires close oversight as to exactly which IPRs must be licensed, how much detailed information must be transferred, how timely updated information must be provided, and how long the duty to license IPRs or supply information will need to last.[307] The European Commission and Microsoft, for instance, disagreed about how much disclosure was required.[308] Disclosure of interface information under the U.S. consent decree has not accom-

[302] *See* 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. TELECOMM. & TECH. L. REV. 77, 83 (2007) (discussing this "forward-looking" approach). Some states that also sued Microsoft for antitrust violations had sought more comprehensive disclosure of interface information, *id.* at 84–87, but the judge overseeing these cases rejected the more expansive interface disclosure request. *Id.* at 103–08.

[303] *Id.* at 83.

[304] Jonathan Zittrain, *The Un-*Microsoft *Un-Remedy: Law Can Prevent the Problem That It Can't Patch Later*, 31 CONN. L. REV. 1361, 1371 (1999).

[305] *Id.* at 1372.

[306] New York v. Microsoft Corp., 224 F. Supp. 2d 76, 171–72 (D.D.C. 2002).

[307] Kovacic, *supra* note 301, at 1304.

[308] *See supra* notes 296–98.

plished its intended objective of bringing about more competition in the PC-based OS market.[309]

While remedies in the *United States v. Microsoft Corp.* case were being hotly debated, some commentators were skeptical that requiring Microsoft to disclose interface information would reduce its dominance in the OS market; in retrospect, this skepticism seems warranted. Some argued instead for a structural remedy, such as breaking Microsoft up into one firm 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.[310] Among the difficulties with this proposal was that it would require legal authorities to make difficult judgments about what "belongs" in an OS and what "belongs" in applications, which they lacked expertise to assess.[311]

Alternatively, some favored breaking Microsoft up into multiple companies ("Baby Bills"), each of which would develop Windows OS technology and license interfaces to applications providers. This might have produced more competition in the OS market, but it risked fragmentation of the OS, which seemed likely to lead to greater development costs and a loss of benefits to consumers and applications developers of a single de facto interface standard.[312]

Yet another alternative for resolving conflicts over IPRs in and disclosures of interface information was invented in the mid-1980s 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.[313] Fujitsu had sold IBM-compatible OS software for mainframe computers without objection from IBM from the mid-1970s until 1982.[314] Then

[309] *See, e.g.*, Page & Childers, *supra* note 302, at 83–84, 126–36.

[310] *See, e.g.*, Kovacic, *supra* note 301, at 1294–304 (discussing proposed structural remedies).

[311] Zittrain, *supra* note 304, at 1370–71.

[312] *See, e.g.*, Stan J. Liebowitz, *Breaking Windows: Estimating Some Costs of Breaking up Microsoft Windows*, 32 UWLA L. REV. 137, 139–40 (2001). *But see* Robert J. Levinson et al., *The Flawed Fragmentation Critique of Structural Remedies in the Microsoft Case*, 46 ANTITRUST BULL. 135, 135–38 (2001); *see also* STAN J. LIEBOWITZ, A FOOL'S PARADISE: THE WINDOWS WORLD AFTER A FORCED BREAKUP OF MICROSOFT (2000) (responding to Levinson et al.).

[313] See Stork, *supra* note 44, at 242–47, for a more detailed explanation of the background of the IBM-Fujitsu dispute and arbitration.

[314] *Id.* at 243.

IBM charged Fujitsu with having misappropriated IP rights in its OS.[315] Fujitsu asserted that it had only appropriated public domain and unprotectable elements from IBM's programs.[316] Although IBM and Fujitsu settled this first dispute in 1983, key terms were left undefined and the compromise soon broke down.[317] Arbitration ensued.[318] One of the difficulties confronting the arbitrators was that the scope of copyright protection in computer programs was unclear at that time.[319] 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.[320]

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 "secured facility" operated by a special set of Fujitsu employees.[321] Fujitsu's "clean-room" team would then analyze the source code and extract interface information.[322] 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.[323] 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.[324]

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

[315] *Id.*

[316] *Id.* at 243–44.

[317] *Id.* at 244–45.

[318] *Id.* at 245.

[319] *Id.*

[320] *Id.* at 246.

[321] *Id.*; *see also* Robert H. Mnookin, *Creating Value Through Process Design: The IBM-Fujitsu Arbitration*, 47 ARB. J. 6, 9–10 (Sept. 1992).

[322] *See* Stork, *supra* note 44, at 246.

[323] *See* Mnookin, *supra* note 321, at 10.

[324] *Id.* at 10. 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 serious disputes, as the parties had adequate incentives to cooperate with this procedure. *Id.* at 11.

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.[325]

E. Promoting Private Ordering as to Interface Patents

Because many ICT industry participants are aware of the high exclusionary potency of interface patents, several private sector initiatives have focused on development of policies to ensure that patents on ICT interfaces will be exercised so as to promote interoperability rather than to thwart it. This is especially important when an interface technique, such as a communications protocol, is under consideration for formal adoption as a standard.

One noteworthy initiative has been the 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.[326] The initial impetus for adoption of this policy was a claim that the W3C's Platform for Privacy Preferences (P3P) standard infringed a non-W3C-member's patent.[327] 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 extensive 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.[328]

The Organization for the Advancement of Structured Information Standards (OASIS) does not mandate RF licensing of

[325] *See* Page & Childers, *supra* note 302, at 112–26 (discussing many difficulties encountered in determining how much information Microsoft was obliged to disclose under the consent decree).

[326] W3C Patent Policy Working Group, W3C Patent Policy §§ 3.1, 5 (Daniel J. Weitzner ed., 2004), 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.* § 4.

[327] *See* Daniel J. Weitzner, Standards, Patents and the Dynamics of Innovation on the World Wide Web pt. II (Nov. 1, 2004), http://www.w3.org/2004/10/patents-standards-innovation.html.

[328] *Id.* at pt. III.

essential interface patents owned by member firms which become OASIS standards.[329] OASIS provides, instead, several licensing options. OASIS seems to have been influenced by the W3C policy, for it developed more than one RF licensing option for TCs operating under OASIS' aegis; yet, it also allows TCs to adopt policies that commit holders of patented technologies adopted as standards to license them on RAND terms.[330] 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 web services standards approved by OASIS.[331] Patents on interface components of OASIS standards are, therefore, generally available on RF 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 to W3C and similar RF policies, on the ground that these licenses include some restrictions that are incompatible with the practices of this community.[332]

The overwhelming majority of SSOs that adopt standards affecting the ICT industry require members who participate in standard-setting processes to disclose any patents they own that are essential to any standard under consideration by that SSO.[333] Most also require a pre-commitment to licensing such

[329] *See* OASIS, OASIS INTELLECTUAL PROPERTY RIGHTS POLICY § 4 (2008), http://www.oasis-open.org/who/intellectualproperty-2008-05-02.pdf (describing three IPR modes including RAND, RF on RAND Terms, and RF on Limited Terms).

[330] *See id.*

[331] Interview with Robert J. Glushko, OASIS Board Member, in Berkeley, Cal. (Apr. 22, 2008).

[332] *See, e.g.*, FREE SOFTWARE FOUNDATION, FSF'S POSITION ON W3 CONSORTIUM "ROYALTY-FREE" PATENT POLICY (2003), http://www.gnu.org/philosophy/w3c-patent.html (expressing objection to field-of-use restrictions that limit practice of the patented idea to those precisely specified as set forth in the license).

[333] Mark A. Lemley, *Intellectual Property Rights and Standard-Setting Organizations*, 90 CAL. L. REV. 1889, 1904–05 (2002) (implicitly requiring disclosure of all essential patents through requirement to disclose all patents known).

patents on RAND terms.[334] So interface patents that have been adopted as standards will generally be available under RAND licenses, even if not on RF terms.[335]

Another private sector 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.[336] 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."[337] The OIN pool acquires software patents of all kinds, including some that cover valuable interfaces.[338] Other similar pools seem to be forming.[339] Some firms are, moreover, making unilateral commitments not to enforce certain interface patents.[340]

[334] *Id.* at 1906. However, there are sometimes disputes about what "reasonable" means. *See id.* (describing a lack of guidance to what "reasonable and nondiscriminatory licensing" terms actually mean).

[335] The Internet Engineering Task Force (IETF) and the European Telecommunications Standards Institute (ETSI) have IP policies that require essential interface standards to be made available on RAND terms. *See* SCOTT BRIM, GUIDELINES FOR WORKING GROUPS ON INTELLECTUAL PROPERTY ISSUES § 5.6 (2004), http://www.ietf.org/rfc/rfc3669.txt (describing IETF's goal of helping a working group choose a technology and offering guidelines for the process including proposed licensing terms); ETSI, *ETSI Rules of Procedure: Annex 6 ETSI Intellectual Property Rights Policy*, *in* ETSI DIRECTIVES 34, 37 (2009), http://portal.etsi.org/directives/25_directives_jan_2009.pdf. ETSI defines ESSENTIAL as: 'ESSENTIAL' as applied to IPR means that it is not possible on technical (but not commercial) grounds, taking into account normal technical practice and the state of the art generally available at the time of standardization, to make, sell, lease, otherwise dispose of, repair, use or operate EQUIPMENT or METHODS which comply with the STANDARD without infringing that IPR. ETSI IPR Policy FAQs, http://www.etsi.org/website/AboutETSI/LegalAspects/IPR_Policy_FAQ.aspx (last visited Apr. 29, 2009).

[336] *See* Open Invention Network, About OIN, http://www.openinventionnetwork.com/about.php (last visited Apr. 29, 2009); Open Invention Network, Members, http://www.openinventionnetwork.com/about_members.php (last visited Apr. 29, 2009).

[337] *See* Open Invention Network's Patent License Agreement, http://www.openinventionnetwork.com/pat_license.php (last visited Apr. 29, 2009).

[338] The Meltzer patent, discussed *supra* note 107, for example, is part of the OIN pool. *See* Open Invention Network's Currently Owned Patents, http://www.openinventionnetwork.com/pat_owned.php (last visited Apr. 29, 2009).

[339] *See, e.g.*, François Lévêque & Yann Ménière, *Copyrights Versus Patents: The Open Source Software Legal Battle*, 4 REV. ECON. RES. ON COPYRIGHT ISSUES 27, 42–43 (2007); Hank Barry, Attorney, Howard Rice Nemerovski Canady Falk & Rabkin, Presentation at the Berkeley Center for Law &

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.[341] Some interface patents are within 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?

Anyone who reviews the extensive literature discussing policy options for responding to the exclusionary potency of ICT interface patents would get the impression that patents on interfaces must be very serious impediments to interoperability. Part II has shown that this literature offers more than twenty-five suggestions for regulating interface patents in order to achieve more compatibility among ICT systems. Within this framing, the only question seems to be which of the many options would best accomplish 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 prevalent, even if not ubiquit-

Technology Symposium: Intellectual Property and Entrepreneurship Conference (Mar. 8, 2008).

[340] Lévêque & Ménière, *supra* note 339. Sun and Microsoft, for instance, have cross-licensed their patent portfolios. *See, e.g.*, Press Release, Microsoft Corp., Microsoft and Sun Microsystems Enter Broad Cooperation Agreement: Settle Outstanding Litigation (Apr. 2, 2004), http://www.microsoft.com/presspass/press/2004/apr04/04-02SunAgreementPR.mspx. Sun could, therefore, have lawfully implemented the Active Directory protocols at issue in the European Commission's action against Microsoft without worrying about patent infringement; the problem was that Sun did not have access to the relevant interface information, not that Microsoft had refused to license the patents. E-mail from David Heiner, Vice President & Deputy Gen. Counsel, Microsoft Corp., to the author (May 13, 2008) (on file with the author).

[341] *See, e.g.*, Parchomovsky & Wagner, *supra* note 95, at 35–36 & n.123, 73–74; Robert P. Merges, *IP Rights and Technological Platforms* (Berkeley Ctr. for Law & Tech., Working Paper No. 64, 2008), http://repositories.cdlib.org/bclt/lts/64/ (proposing that many market factors provide incentives for interoperability and arguing that *ex post* measures are superior to *ex ante* regulations).

ous.[342] Market incentives, driven in large part by the potential for generating network effects, are often strong enough to induce many firms either to publish ICT interfaces without IP restrictions or to license interface information on reasonable terms. Many interface patents are licensed through cross-licensing arrangements common in ICT industries. Insofar as SSOs have adopted patented interfaces as standards, licenses are likely to be available under either RF or RAND terms.[343] The more fundamental the interfaces are to the functioning of key infrastructures, such as the World Wide Web and web services, the more likely the patents are to be available on RF terms.[344]

This is not to say that patents on interface designs never impede interoperability. Nintendo was, after all, able to block Atari Games from making and selling games for its NES console because of the patent it obtained for an authentication mechanism.[345] That case was, however, decided fifteen years ago.[346] Very few reported cases since then have involved ICT interface patents whose exercise was impeding interoperability.[347]

[342] *See, e.g.*, GASSER & PALFREY, *supra* note 1, at 5–9.

[343] *But see* Lemley & Shapiro, *supra* note 93, at 2042–43 (suggesting that royalty caps or "step-down" royalties are better alternatives to RF and RAND licensing arrangements but suffer from raising "antitrust flags").

[344] Owners of patents on interface techniques may, of course, not be members of the W3C, OASIS, or other SSOs, and hence not committed to the RF or RAND policies of those organizations. *See supra* notes 332–39 and accompanying text. Private initiatives to foster interoperability through SSO policies may thus 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. *See* eBay v. MercExchange, L.L.C., 547 U.S. 388, 396–97 (2006) (Kennedy, J., concurring) (explaining that in similar cases "legal damages may well be sufficient to compensate for the infringement and an injunction may not serve the public interest").

[345] *See supra* Part I.D.

[346] Atari Games Corp. v. Nintendo of Am., Inc., 975 F.2d 832, 832 (Fed. Cir. 1992).

[347] It is possible that some firms have given up on ICT development projects because they were unable to license interface patents; it is, however, impossible to know how often this has happened. Litigation is an imperfect indicator of patents as impediments to interoperability. However, I expected to find more cases in which patents were being asserted to thwart compatibility

Some anecdotal evidence exists as to open source software projects that did not go forward because of interface patents.[348] Yet, even open source projects have been able to implement some patented interfaces. Samba and Microsoft, for example, were able to reach an agreement on Samba's use of certain network communications protocols.[349] (Of course, Microsoft's willingness to license these protocols to Samba on GPL-friendly terms was due in large part to the Commission's oversight of Microsoft's interface licensing practices.[350]) It is a fair inference that organizations, such as W3C, OASIS, and METI, would not have undertaken their policy initiatives to blunt the exclusionary potency of interface patents if these organizations thought such patents would never or only rarely present impediments to interoperability.

Nor is it to say that patents will not become a bigger impediment to interoperability in the future. FFII has expressed concern that Microsoft will undo the European Commission's ruling by seeking ever more patents on interfaces.[351] Yet, even if Microsoft gets more patents on interface techniques, there is no reason to believe that Commission will be any more deferential to those patents than it was to the patents raised in the 2004 proceeding. Microsoft was not able to persuade the CFI that courts should be more deferential to a firm's refusal to license technological IPRs, such as patents on interfaces, than to refusals to license copyrights.[352] The Commission and the CFI

because would-be interoperators would likely challenge the validity of interface patents in hopes of using the techniques if the patentee was either refusing to license the patents or demanding unreasonable terms.

[348] *See, e.g.*, Andy Tai, *Microsoft Prohibits GPLed Work Via Licensing of CIFS Standards*, ADVOGATO, Apr. 5, 2002, http://www.advogato.org/article/453.html.

[349] *See, e.g.*, Microsoft and Samba Finally Come to Terms Over Windows Protocols, http://blogs.zdnet.com/microsoft/?p=1064 (Dec. 20, 2007, 12:49 PST). Microsoft was initially 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; Microsoft objected to this. However, the final agreement allowed Samba to use the GPL for the software implementing the Microsoft patented protocols. *Id.*

[350] *See* Page & Childers, *supra* note 91, at 333 (characterizing the Microsoft-Samba license as "by far the most important tangible outcome of the European *Microsoft* case").

[351] *See* Press Release, Foundation for a Free Information Infrastructure, Microsoft Will Trump EU Competion Ruling with Patents (Sept. 17, 2007), http://press.ffii.org/Press_releases/Microsoft_will_trump_EU_competition_ruling_with_patents (describing recent patents obtained by Microsoft).

[352] Case T-201/04, Microsoft Corp. v. Comm'n, 2007 E.C.R. II-3601, ¶ 693

regard competition law as an appropriate means to monitor a dominant party's use of IP to impede interoperability. In the United States, courts are more likely to evolve doctrines within patent law to respond, as needed, if patents are impeding socially desirable interoperability.

The *eBay* decision provides U.S. courts with a basis for awarding reasonable royalties to owners of patents instead of injunctions for interfaces essential to interoperability.[353] This approach may be more likely if the patent would block access to widely used infrastructures,[354] although it is noteworthy that the Federal Circuit was inhospitable to Vonage's argument that an injunction should not issue to block its use of Verizon's patented interface because of the impacts of the injunction for millions of users of its VoIP services.[355] In the post-*eBay* caselaw thus far, injunctive relief has generally been withheld only in cases that appear to involve "patent trolls."[356]

It is, however, erroneous to construe *eBay* as confining judicial discretion to withhold injunctive relief to patent-troll-like cases. *eBay* held that courts should carefully consider in all IP cases the same four factors traditionally used in making judgments about whether to order injunctive relief.[357] Nokia's amicus brief in *eBay* made a strong argument about why injunctive relief should be withheld in cases involving patents on interfaces essential to interoperability.[358] Courts should be es-

(refusing Microsoft's argument without the presence of the exceptional circumstances identified in *Magill* and *IMS Health*).

[353] *See supra* notes 212–26 and accompanying text.

[354] *See supra* Part II.C.

[355] *See* Verizon Servs. Co. v. Vonage Holdings Co., 503 F.3d 1295, 1310–11 (Fed. Cir. 2007) ("[A]ffirming the district court's decision to issue an injunction . . . .").

[356] *See* 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, 312–22 (2007). Courts may be wary of awarding damages in lieu of injunctions because of the difficulties that typically attend price-setting by nonmarket actors through compulsory licenses, especially as applied to developers of competing platforms. *Id.* at 342–43. If an interface patentee deliberately stays outside of a standard setting process that is likely to implicate its IPRs in order to avoid a RAND commitment, courts might find this an appropriate case in which to award damages but not injunctive relief, as this may be seen as troll-like behavior.

[357] eBay Inc. v. MercExchange, L.L.C., 547 U.S. 388, 391 (2006) ("[A] plaintiff seeking a permanent injunction must satisfy a four-factor test before a court may grant such relief.").

[358] Brief of Nokia Co. as Amici Curiae Supporting Petitioners at *2, eBay Inc. v. MercExchange, L.L.C., 547 U.S. 388 (2006) (No. 05-130), 2006 WL

pecially receptive to a damage-based remedy if an interface patent covers one small component of a complex multi-component technology so that, in effect, an injunction would result in thwarting interoperability as to components not covered by the patent.[359] It is also conceivable, although somewhat less likely, that U.S. courts will adopt the fair use defense proposal discussed above to balance the patent holders' and public interests at stake in cases involving patents affecting interoperability.

U.S. courts are, however, unlikely to rule that a firm's refusal to license an interface patent constitutes patent misuse, even if the claim was wrapped in the language of tying arrangements.[360] The U.S. Patent Act specifically provides that refusing to license patents is not misuse.[361] Nor are U.S. courts likely to 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.[362] In the aftermath of *Trinko*, it is doubtful that U.S. courts would regard ICT interfaces as essential facilities, refused

235005 ("[T]he decision below threatens to thwart the innovative and procompetitive work of standard-setting organizations.").

[359] General Electric, for instance, claimed patent rights in a method and system for transferring images from an ultrasound system to third-party storage and printing systems. *See* U.S. Patent No. 6,210,327 (filed Apr. 3, 2001). Use of that method and system was essential for commercial viability of any business of making and selling diagnostic ultrasound machines. GE sued Sonosite for infringing this patent, although it eventually dropped this claim. Interview with Ted Sichelman, Research Fellow, Berkeley Center for Law & Technology, in Berkeley, Cal. (Aug. 22, 2008).

[360] *See supra* notes 191–93 and accompanying text for Prof. Cohen's argument as to why the Nintendo lock-out system was patent misuse. *But see*, *In re* Indep. Serv. Org. Antitrust Litig., 203 F.3d 1322, 1328 (Fed. Cir. 2000) (holding that a patentee can lawfully refuse to license third parties who want to make replacement parts). One might argue that treating lock-outs as misuse is less likely after enactment of the anticircumvention rules of the Digital Millennium Copyright Act because it privileges use of technical protection measures (TPMs) to protect copyrighted works. 17 U.S.C. § 1201 (2006); *see* Randal C. Picker, *Copyright and the DMCA: Market Locks and Technological Contracts*, *in* ANTITRUST, PATENTS AND COPYRIGHTS, *supra* note 297, at 181 (noting how the DMCA violates content locks). Yet, some cases have interpreted the anticircumvention rules narrowly to preclude lock-out TPMs that do not cause copyright infringements. *See, e.g.*, Chamberlain Group, Inc. v. Skylink Tech., Inc., 381 F.3d 1178, 1202 (Fed. Cir. 2004) ("We conclude that 17 U.S.C. § 1201 prohibits only forms of access that bear a reasonable relationship to the protections that the Copyright Act otherwise affords copyright owners.").

[361] 35 U.S.C. § 271(d)(5) (2006).

[362] *See supra* notes 300–03 and accompanying text.

access to which violates U.S. antitrust law.[363] Of course, a firm that has otherwise violated the antitrust laws may, as in *United States v. Microsoft*, be required to license patents and provide interface information as part of the remedy.[364]

This Article endorses withholding injunctive relief in an appropriate case in which an interface patent is essential to achieving interoperability and compensation would allow the patentee to recoup its R&D expenses; it also endorses some patent reform measures discussed in Part II.B. There is, however, at this time too little empirical evidence that patents are such a major impediment to interoperability to justify adoption of stronger policy measures,[365] such as excluding interfaces from patentable subject matter or immunizing the use of patented

[363] It is certainly possible that U.S. antitrust and patent misuse law may evolve to regulate interface patents in the future if such patents emerge as substantial impediments to interoperability and threaten the viability of competition in the ICT industry. U.S. courts have given relatively little consideration to the power of network effects in ICT systems and how this interacts with proprietary control over interfaces. If U.S. patent and antitrust officials 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. U.S. courts are likely, however, to be more concerned than the Commission and CFI were to the chill on incentives to invest in innovation that forced disclosure of interfaces and licensing of patents may bring about. *See, e.g.*, Verizon Commc'ns Inc. v. Law Offices of Curtis V. Trinko, LLP, 540 U.S. 398, 407–08 (2004) ("Compelling such firms to share the source of their advantage is in some tension with the underlying purpose of antitrust law, since it may lessen the incentive for the monopolist, the rival, or both to invest in those economically beneficial facilities.").

[364] *See supra* Part II.D.

[365] I recognize the dangers of embracing the null hypothesis. Yet, consider the paucity of litigation about interface patents and even of anecdotal evidence of interface patents as impediments to interoperability. I spoke at length with ICT industry experts from small and large firms expecting to find many examples of patents as impediments to interoperability and was surprised at how few emerged. Widespread cross licensing in the IT industry, RF and RAND commitments for patented standards, and many examples of successful licensing of interface patents support my conclusion that the evidence at present of patents as significant impediments to interoperability is relatively weak. *See* Ronald J. Mann, *Commercializing Open Source Software: Do Property Rights Still Matter?*, 20 HARV. J.L. & TECH. 1, 10 (2006) (discussing the "widespread use of cross licensing"). It is also worth noting that there is not just one software industry, but many software industries focused on different market sectors, and many software sector markets are relatively small. While an interface patent may be asserted to thwart interoperability in some niche markets, the larger or wider the software market, the more likely it is that cross licensing or SSO policies will blunt the exclusionary potency of interface patents. More systematic empirical research might, of course, identify problems arising from interface patents, such as allowing patentees to extract royalties out of proportion to the value of the innovation.

interfaces to achieve compatibility.[366] While there are good reasons to question the suitability of patents as a form of IP protection for computer program innovations,[367] insofar as patents *are* available for software inventions, it is difficult to justify withholding them from interface techniques, as long as they meet patent law's substantive standards. Interfaces enable the development of new features of ICT systems and are inextricably connected to the innovative features they enable.[368]

Even though some interface designs are inventive and costly to develop, 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. Patent incentives may not actually be necessary to bring interfaces into existence.[369] Developing software without interfaces is like building a house without windows or doors.[370] Although it can be done, there is little point in doing it, as ICT technologies are generally more useful when they in-

[366] Excluding interfaces from patent subject matter would likely be difficult to monitor, as firms would almost certainly encourage their lawyers to engage in artful patent drafting to claim rights in interfaces without designating them as such. An immunity from infringement provision as to techniques essential to interoperability would be more likely to achieve its goal. However, as noted above, such an immunity provision would risk running afoul of TRIPs obligations. *See supra* notes 188–89.

[367] *See* Samuelson, Benson *Revisited*, *supra* note 45, at 1031–32 ("If the software industry neither wants nor needs the patent system in order to be a vital and innovative industry, then, as a matter of public policy, it is sensible not to use the patent system for the protection of program-related innovations."); Samuelson et al., *supra* note 146, at 2343–47 ("The dual character of computer programs—which are both writings and machines at the same time—has presented some difficulties for those wanting to use patent law as a means of legal protection for software innovations.").

[368] Thanks to Robert Barr for making this point. Barr also pointed out that if engineers at SSO technical meetings spend hours debating which is 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.

[369] *See, e.g.*, O'Rourke, *supra* note 96, at 1216 ("[B]ecause significant first-mover advantages and returns larger than those of conventional markets often characterize network markets, investors may not need the inducement of intellectual property rights at all to encourage them to fund innovation."). As noted in Part II.A, a decision to exclude interfaces from patenting would likely lead to greater reliance on trade secrecy protection. Such a rule would also be difficult to monitor effectively since a claim may cover an interface component without the application specifically calling it that. Interface innovations are, moreover, likely to be tied to feature innovations, and claimed as such.

[370] *See* David Orenstein, *QuickStudy: Application Programming Interface (API)*, COMPUTERWORLD, Jan. 10, 2000, http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=43487 (making this analogy).

teroperate with other entities.[371] Firms such as Microsoft are unlikely to cease developing interfaces or improving interfaces if they cannot patent them. The EU could reasonably have decided that software developers would have sufficient incentives to develop programs by the expansive protection the Directive provides to software internals (other than interfaces) and by outlawing decompilation except when motivated by compatibility considerations. Such expansive protection could provide adequate incentives for software development even if without protection for interfaces (which, after all, can still be kept secret).

Although the Supreme Court may revisit the patentability of software and narrow its scope,[372] this would have little impact on the patentability of interfaces insofar as they are tied to enabling product features. The bigger challenge in the United States, is weeding out patents on arbitrary or trivial interface designs. This may be accomplished through a reinvigoration of the invention standard after the Supreme Court's *KSR* decision and other patent reforms, such as an improved post-grant review regime.[373]

While nations should not require interface techniques to be any more innovative than other patentable designs, patent officials would be well advised to examine claims for patents on interface techniques more rigorously than they examine other kinds of claims and perhaps to require more meaningful disclosure of interface techniques if substantive standards of patentability are met. This greater scrutiny would be wise given that firms have incentives to patent interface techniques less because of any innovation they might or might not embody and more because of their exclusionary potency.

There are certainly some sectors in which there is less interoperability than one might wish for (e.g., digital music services),[374] but patents do not seem to be the principal impediments to interoperability in those sectors. Firms that choose

[371] *Id.*

[372] *See* Pamela Samuelson, *Revisiting Patentable Subject Matter*, 51 COMM. ACM, July 2008, at 20, 22 ("If . . . the Federal Circuit reverses the BPAI's ruling in *Bilski* . . . the Supreme Court will probably review the *Bilski* case to clarify what standards the PTO and Federal Circuit should apply in judging which processes are eligible for patent protection.").

[373] See supra Part II.B.

[374] France has undertaken regulatory efforts to bring about greater interoperability as to digital music. *See* L.C. Angell, *French iTunes Interoperability Law Goes Into Effect*, ILOUNGE, Aug. 3, 2006, http://www.ilounge.com/index.php/news/comments/french-itunes-interoperability-law-goes-into-effect/.

noninteroperable business strategies, as Apple did with iTunes,[375] 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), seem to be able to rely on trade secrecy and their ability to re-engineer and re-implement their interfaces if a competitor successfully reverse engineers and implements an earlier interface. Patents may not be necessary for firms to succeed with business models built on noninteroperable or controlled interoperability strategies.

Lack of access to interface information is often a greater impediment to interoperability than patents are. Should nations consider mandating disclosure interface information in order to foster more interoperability? The EU Software Directive did not go so far. Drafters of the Directive may have believed they did enough to induce disclosure of interface information by authorizing reverse engineering for purposes of achieving interoperability.[376] They may not have anticipated the development of very large and complex systems, such as the Windows OS, as to which reverse engineering has become an infeasible way to get access to interface information.[377] The Software Directive provides no remedy to would-be reverse engineers if interface information cannot be obtained by reverse engineering.[378] Even when reverse engineering does enable second comers to obtain interface information, there may be no legal recourse if the developer of the interface tweaks the interface and thereby thwarts unlicensed firms from attaining compatibility for more than a short time.[379]

[375] *See* Scott M. Fulton, III, *EU Consumer Protection Targets iTunes*, BETANEWS, Mar. 12, 2007, http://www.betanews.com/article/EU-Consumer-Protection-Targets-iTunes/1173719408 (noting how officials in Norway challenged the legality of iTunes, in part because of its noninteroperability).

[376] Software Directive, *supra* note 5, art. 6(1).

[377] Case T-201/04, Microsoft Corp. v. Comm'n, 2007 E.C.R. II-3601 ¶ 362 (stating that reverse engineering was not a feasible way to get WGS-OS interface information).

[378] *See* Software Directive, *supra* note 5, art. 6 (providing protection only to the extent a program can be reverse engineered).

[379] *See, e.g.*, Bill Rosenblatt, *RealNetworks and Motorola Open Apple iTunes/iPod Stack*, DRM WATCH, July 28, 2004, http://www.drmwatch.com/drmtech/article.php/3387481 (noting how RealNetworks reproduced the file format design of iTunes). Thereafter, Apple changed its interface and undid RealNetworks temporary interoperability advantage. Jon Borland, *Apple Fights RealNetworks' 'Hacker Tactics'*, ZDNET ASIA, Dec. 15, 2004, http://www.zdnetasia.com/news/security/0,39044215,39209366,00.htm.

Nations could, of course, require developers to be more forthcoming about interfaces, either as a *sui generis* matter or as a precondition of getting patent protection for interface techniques. Firms often prefer to maintain interface information as a secret, but nations can mandate disclosure of information if they deem it necessary to achieve public policy goals.[380] Nations could also regulate those who alter their interfaces for the purpose of making a competitor's product noninteroperable.[381]

Before mandating disclosure or outlawing "predatory" changes to interfaces, nations should reflect on the likely effectiveness of such regulation. Consider, first, the problems likely to attend any effort to regulate interface changes. Firms frequently change the interfaces of their technologies when they add new features to their products.[382] Consumers typically benefit when new features are made available; it would be unwise to discourage firms from evolving their products. Regulators may be hard pressed to determine when firms have changed interfaces to facilitate new features and when they did so to harm competition. Consider also that innovators may sometimes regard making changes to interfaces as an appropriate self-help response to competitors who are trying to free-ride on their platform's success, as Apple did after RealNetworks reverse-engineered iTunes interfaces to make its media player compatible with Apple's system.[383]

[380] *See* Page & Childers, *supra* note 91, at 351 ("The EC has required Microsoft to make disclosures sufficient to allow rivals to create server software functionally equivalent to Microsoft . . . ."). It is as yet unclear whether France's effort to regulate disclosure of interface information for digital music systems will succeed in promoting greater interoperability in the digital music sector.

[381] Some commentators have endorsed the use of antitrust law to patrol what they call "predatory innovation." *See, e.g.*, Montagnani, *supra* note 239, at 306–08; van Rooijen, *supra* note 281, at 82–83.

[382] van Rooijen, *supra* note 144, at 136 ("[C]hange seems to be a dangerous ground for antitrust liability. Innovation necessarily involves change."). It is important to understand that firms often have incentives to provide "backward compatibility" so that software and documents developed for a previous iteration of the platform can continue to interoperate with later iterations of the platform. Firms are most likely to change their interfaces when they give up on backward compatibility by releasing a new version of the product (e.g., version 3.0 may have new interfaces, though versions 2.3, 2.4, and 2.5 will have the same ones).

[383] *See* Borland, *supra* note 379. Apple might well have considered RealNetworks as the predator and itself as the prey in this situation. Who is the predator and who is the prey is, thus, not all that straightforward and may well be contestable.

Mandating disclosure of interface information may initially seem a simple solution, but in practice, it too may be fraught with difficulties. The European Commission's effort to oversee Microsoft's compliance with the order to disclose interface information is a cautionary tale.[384] Microsoft produced more than thirty thousand pages of technical information;[385] yet the Commission deemed the disclosures insufficient and heavily fined Microsoft for withholding key information.[386] Determining which components of an ICT product, especially computer programs, are functioning as an interface or are essential to interoperability may be technically very complicated and hotly contested.

Achieving interoperability is, moreover, often a more complex technical goal than legal commentators and policymakers have sometimes acknowledged. Even when one has plentiful access to APIs and protocols, it can still be very difficult to achieve interoperability.[387] To make an ICT system interoperable requires fine-grained agreements on syntax and semantics.[388] Ambiguities in interface specifications often make it difficult to implement APIs.[389] Even when interface designs are

[384] The U.S.-forced disclosure of interface information program ran into similar difficulties. *See* Page & Childers, *supra* note 91, at 337–39.

[385] Castle, *supra* note 290.

[386] Commission Decision, *supra* note 6, ¶¶ 728, 731, 1078.

[387] 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 in representation of the content 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 8, § 6.0.

[388] *Id.* § 6.0. If, for example, an interface calls for location information, interoperability may not occur if the platform is expecting GPS coordinates and the application has encoded location information in postal codes. Similarly, an API for requesting weather information may be difficult to implement 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 great deal.

[389] Open interfaces are generally more programmer-friendly than proprietary ones because programmers who implement open interfaces are more

well-documented and standardized, firms may still need to develop guidelines, obtain expert advice about how to comply with them, and even to obtain access to test suites and other tools to get the details right.[390]

It may thus be unwise to mandate full disclosure of interface information across the board for at least three reasons. First, many technical difficulties attend attaining interoperability in the real world, even when one has considerable amounts of interoperability information. Second, there is considerable imprecision about exactly what information would have to be disclosed to achieve interoperability. Third, there is a risk that requiring full disclosure of all minute details necessary to achieve interoperability will undermine incentives to invest in innovative ICT products.

Nations could, however, facilitate greater access to interface information through some more modest measures. They 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 undertaken for purposes of discerning information essential to interoperability.[391] Nations might also recognize or create a privilege in patent law, akin to that already recognized in copyright, anticircumvention, and trade secrecy laws, to permit reverse engineering for purposes of getting access to interface information.[392]

There is finally the question about whether refusals to disclose or 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 product or service.

The principal argument for allowing unlicensed second-comers to use interface information to develop complementary products, but not functional equivalents (or in the Commission's terms, one-way, but not two-way interoperability) is that

likely to refine the specification to provide more precise substitutes for opaque or confusing terms. Erik Wilde, *What Are You Talking About?*, *in* 2007 IEEE INTERNATIONAL CONFERENCE ON SERVICES COMPUTING 256, 258 (2007).

[390] *See, e.g.*, Page & Childers, *supra* note 302, at 130 (explaining that Microsoft was required to undertake the development of test suites).

[391] Software Directive, *supra* note 5, art. 9.1.

[392] *See supra* Part II.B. For a discussion of the interoperability exception to the anticircumvention rules, see Aaron K. Perzanowski, *Rethinking Anti-Circumvention's Interoperability Policy*, 40 U.C. DAVIS L. REV. (forthcoming 2009).

there is a risk that the developer of the interface will not be able to recoup its R&D expenses because functional equivalents will undercut sales of that developers' principal products (e.g., the platform), whereas complements are likely to build demand for the platform.[393] Establishing a reasonable royalty for licensing interface information as to developers of complementary products will generally be easier than establishing a reasonable royalty for licensing as to developers of functionally equivalent products, as the platform's provider probably has a licensing program for development of complementary products, but probably not for functional equivalents.

However, there are some reasons why IP law should treat interfaces the same, regardless of whether they are intended for use in building complements or equivalents. For one thing, in today's complex networked world, it is no longer as easy as it once was to distinguish between complements and substitutes. Interfaces now enable more complex exchanges than before; a network-based program may, for example, be acting as a complement one moment, but as a functional equivalent the next.[394] Any effort to make a sharp legal distinction between interface information for complements and functional equivalents is thus likely to break down.[395] Second, much of the information necessary to make a complementary product is also necessary to know to make a functional equivalent, and vice versa. This too makes a sharp legal distinction between them problematic. Third, it is rare in IP law to treat information as protectable for one purpose, but not for another.[396] Neither copyright nor trade

[393] *See* Weiser, *supra* note 30, at 560–61 (arguing that IP law should allow unlicensed parties to develop complements, but not substitutes, at least not until and unless the incumbent becomes a monopolist). The ability to recoup investments was the main concern Microsoft raised to the Commission about being forced to supply highly detailed interface information to Sun. Commission Decision, *supra* note 6, ¶ 274.

[394] *See, e.g.*, Lévêque, *supra* note 247, at 112–13 (describing how servers and PC OS are partial complements and partial substitutes).

[395] Consider this variant on *Altai*. Suppose Altai had reverse engineered the compatibility component of CA-Scheduler to get access to interface information. It would have done so in order to make a program that was a functional equivalent to CA-Scheduler, but it needed the interface information primarily in order to interoperate with the IBM OS programs with which both Altai's and CA's programs were designed to run (that is, to develop a complement to the IBM platform).

[396] *International News Service v. Associated Press* is a rare example in which certain information (news) was treated as private property as between INS and AP, although it was common property as to the general public. 248 U.S. 215, 235 (1918). The *INS* concept of "quasi-property" has been discredited

secrecy law, for example, has distinguished between reverse engineering undertaken to make complements or substitutes when analyzing the lawfulness of reverse engineering and reuse of information discerned from reverse engineering.[397] Reverse-engineers frequently aim to make competing products.[398] Creating a new legal distinction to allow reuse of interface information to make complements, but not substitutes, would be a novel move for IP law. Fourth, firms sometimes adopt business strategies that do not conform to the usual platform/complement story. Nintendo, for example, lost money every time it sold an NES console.[399] Its strategy for achieving financial success depended upon controlling the market for complementary products (i.e., games) for this platform.[400] By selling unlicensed complements for Nintendo's console, Atari Games was arguably thwarting Nintendo's recoupment strategy. Finally, there is some evidence that firms can recoup R&D investments even when they facilitate access to interface information for developers of functionally equivalent products.[401] IBM and Fujitsu, for instance, were able to coexist in supplying functionally equivalent software for mainframe computers. Some competition among different implementations of a platform may be beneficial to consumers who at the same time also benefit from the platform's stability as a de facto standard.

over time. Pamela Samuelson, *Information as Property: Do* Ruckelshaus *and* Carpenter *Signal a Changing Direction in Intellectual Property Law?*, 38 CATH. U. L. REV. 365, 388–95 (1989).

[397] *See* Sega Enters. Ltd. v. Accolade, Inc., 977 F.2d 1510, 1520 (9th Cir. 1992) (agreeing with Accolade that it is fair use to reverse engineer to get information necessary to make complementary products); Sony Computer Entm't, Inc. v. Connectix Corp., 203 F.3d 596, 602 (9th Cir. 2000) (holding that it is fair use to reverse engineer to get access to information to make functional equivalent).

[398] *See* Samuelson & Scotchmer, *supra* note 85, at 1577 (identifying the intention of reverse engineering to "draw customers away from the maker of the reverse-engineered product").

[399] *See id.* at 1618–19 ("In the game system market, platform developers typically lose money on sales of consoles, making up losses on sales of games and peripherals.").

[400] *Id.*

[401] The CFI Decision relied in part on this reasoning in responding to Microsoft's concerns about its ability to recoup R&D expenses if forced to disclose interface information to Sun and others. Case T-201/04, Microsoft Corp. v. Comm'n, 2007 E.C.R. II-3601, ¶ 710. It also relied on Microsoft's disclosure of interface information under the U.S. consent decree and the settlement with Sun. *Id.* ¶¶ 693, 703. *But see* Weiser, *supra* note 30, at 560–68 (expressing concerns about the ability of platform providers to recoup R&D expenses if competitors can freely use interfaces to develop directly competing platforms).

It is, however, noteworthy that the three most contentious disputes over program interfaces—the European Commission's case against IBM in the 1980s, the IBM-Fujitsu litigation/arbitration, and the Commission's recent case against Microsoft—all involved efforts by dominant firms to thwart competitors from making functional equivalent products, not complements.

What we can say with some confidence is that interface patents pose the gravest risks for competition and follow-on innovation when the patents cover techniques that are essential to interoperability, when those patents are held by firms with market power, and when there are incentives for firms in dominant positions to exercise interface patents to exclude competitors from the market or to leverage the firm's power in one market to gain power in an adjacent market, especially disruptive new entrants with an entrepreneurial bent.[402] Any regulatory intervention as to interface patents should only be undertaken in a targeted manner to address specific abuses, at least until there is systematic empirical evidence that patents are, in fact, more serious impediments to interoperability than they seem to be today.

CONCLUSION

Interoperability among ICT components and systems is widely viewed as socially desirable insofar as it promotes competition, innovation, and consumer choice. Because firms often have incentives to reveal information necessary to enable others to build compatible technologies so that their platforms are more attractive to consumers, many either do not assert IP rights in their interfaces or license such rights on open terms. Some firms, however, prefer to maintain interfaces as trade secrets or to seek patents for them as part of a business strategy that is aimed at controlling which firms are able to interoperate with their technologies. The principal advantage of patents over trade secrecy in the protection of interfaces is that patents confer a strong legal right to exclude others from building technologies that incorporate the patented technique. If the technique is essential to achieving interoperability, the exclusionary potency of an interface patent is considerable.

[402] *See* Farrell & Weiser, *supra* note 7, at 105–19 (discussing circumstances in which leveraging power from one market to another may be rational and anticompetitive); O'Rourke, *supra* note 96, at 1216 (leveraging may be a rational strategy in network markets).

Many commentators and policymakers have recognized that patents on interfaces can be and sometimes have been exercised to block the development of interoperable technologies. Out of concern about the exclusionary power of such patents, they have proposed a wide array of legal and policy measures to ensure that interoperability can occur. This Article is the first comprehensive analysis of the twenty-some policy responses that have thus far been proposed or implemented to blunt the power of interface patents.

This Article makes four main points. First, there is less need for strong measures, such as barring patents on interface innovations or treating the exercise of interface patents to block interoperability as misuse of the patents, than some commentators seem to believe. Second, insofar as interface patents do emerge as more serious impediments to interoperability than they have been to date, there are adequate policy responses in place in various countries that can be used to address them. Third, some tailoring of patent rules and patent reforms may be advisable in order to promote greater interoperability among ICT systems. Fourth, patents are often less of an impediment to interoperability than the secrecy of interface information, which may be difficult or impossible to reverse engineer, ambiguities in interface specifications, and changes in interfaces that may be made in new versions or features of an ICT system. This Article explains why it may be difficult to bring about more interoperability by mandating greater disclosure of interface information or regulating what kinds of changes firms can make to their interfaces.