An epitaph for traditional copyright protection of network features of computer software

BY PETER S. MENELL*

Effective control of interface specifications and other network features of modern computer technologies has been the key to market dominance in the information and computer age. It is not surprising, therefore, that intellectual property protection for network features of computer programs has been a central battleground in the competitive struggle. Just a decade ago, copyright law was emerging as a potent mode of protection for network features of computer software. In 1983, the Third Circuit commented, in one of the first major cases regarding the protection of operating system software for microcomputers, that achieving "total compatibility with independently developed application programs . . . is a commercial and competitive objective which does not enter into the somewhat metaphysical issue of whether

* Professor of Law and Co-Director, Berkeley Center for Law & Technology, University of California at Berkeley School of Law.

AUTHOR'S NOTE: *I would like to thank Mark Lemley and Howard Shelanski for their input on this project.*

particular ideas and expressions have merged."[1] A number of lower courts around the country extended copyright protection to the screen displays and other features of computer programs.[2] In 1987, the Third Circuit pronounced a test for determining the scope of copyright protection for computer programs that potentially afforded broad protection under copyright law for the network features of computer systems.[3] A few other courts, most notably the lower court in the Lotus litigation,[4] followed this general approach in determining the scope of protection for user interfaces. Meanwhile, advocates for some of the largest computer companies were advancing the view that copyright should provide even broader protection for network and other functional features of computer software.[5] The risks of developing interoperable systems loomed large.

The picture could not look more different today. Over the course of the past decade, the federal courts have reasserted fundamental limitations on the scope of copyright, effectively excluding network features from the domain of copyright protection. This dramatic turnaround reflects copyright's adaptability to technological change,[6] the role of academic

[1] *See* Apple Computer v. Franklin Computer Corp., 714 F.2d 1240, 1253 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984).

[2] *See, e.g.*, Broderbund Software Inc. v. Unison World, 648 F. Supp. 1127 (N.D.Cal. 1986); Digital Communications Associates, Inc. v. Softklone Distributing Corp., 659 F. Supp. 449 (N.D.Ga. 1987).

[3] *See* Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031 (1987).

[4] *See* Lotus Development Corp. v. Paperback Software Int'l, 740 F. Supp. 37 (1990); Lotus Development Corp. v. Borland Int'l, 799 F. Supp. 203 (1992).

[5] *See* Goldberg & Burleigh, *Copyright Protection for Computer Programs: Is the Sky Falling?*, 17 AIPLA Q.J. 294 (1989); Clapes, Lynch, & Steinberg, *Silicon Epics and Binary Bards: Determining the Proper Scope of Copyright Protection for Computer Programs*, 34 UCLA L. REV. 1493 (1987).

[6] *See* P. GOLDSTEIN, COPYRIGHT'S HIGHWAY: THE LAW AND LORE OF COPYRIGHT FROM GUTENBERG TO THE CELESTIAL JUKEBOX (1994) (illustrating the copyright law's historic adaptability to the challenges of new technology).

scholarship,[7] amicus curiae,[8] and interest group representatives in educating the courts about the proper interpretation of copyright law,[9] and the ability of the federal judiciary to correct false starts

[7] Following the early court decisions expanding the scope of copyright protection, Milton Wessel, the director of the Arizona State University Center for the Study of Law, Science and Technology, along with professors Dennis Karjala, Pamela Samuelson, and myself, organized a conference of leading intellectual property professors to address the proper scope of copyright protection for computer software. *See* Wessel, *Introductory Comment on the Arizona State University LaST Frontier Conference on Copyright Protection of Computer Software*, 30 JURIMETRICS J. 1 (1989). The analysis and conclusions of this conference, contained in a consensus statement published at that time, *see LaST Frontier Conference Report on Copyright Protection of Computer Software, id.* at 15–33 [hereinafter cited as Consensus Statement], has since been reflected in the decisions of the federal courts. In addition, intellectual property professors have filed amicus curiae briefs in many of the leading cases. *See, e.g.,* Brief Amicus Curiae of Professors Peter S. Menell and Dennis S. Karjala in Support of Respondent, Lotus Development Corp. v. Borland Int'l, Inc., U.S. Supreme Court, No. 94-2003 (October Term, 1995); Brief Amicus Curiae of Professors Peter S. Menell and Dennis S. Karjala in Support of Appellant, Lotus Development Corp. v. Borland Int'l, Inc., 1st Circuit Court of Appeals (1993), *reprinted in* 10 HIGH TECH. L.J. 1177 (1995); Brief Amicus Curiae of Copyright Law Professors in Support of Appellant, Lotus Development Corp. v. Borland Int'l, Inc., 1st Circuit Court of Appeals (1993), *reprinted in* 16 HASTINGS COMM. & ENT. L.J. 657 (1994); *Brief Amicus Curiae of Eleven Copyright Law Professors in Sega Enterprises Ltd. v. Accolade, Inc.*, 33 JURIMETRICS J. 147, 155 (1992). The seminal court decisions have drawn directly upon the academic literature in developing the modern tests for copyright protection of computer software. *See* Computer Associates Int'l v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992); Apple Computer, Inc. v. Microsoft Corp., 799 F. Supp. 1006 (1992), *aff'd in part, rev'd in part,* 35 F.2d 1435 (1994), *cert. denied,* 513 U.S. 1184 (1995).

[8] *See, e.g.,* Lotus Dev. Corp. v. Borland Int'l, Inc., U.S. Supreme Court, No. 94-2003 (October Term, 1995), Brief Amicus Curiae of Economics Professors and Scholars, Brief Amicus Curiae of Computer Scientists, Brief Amicus Curiae of American Committee for Interoperable Systems (ACIS).

[9] New interest groups, such as the American Committee for Interoperable Systems (ACIS) (an organization of companies that develop software and hardware products that interoperate with computer systems

and further the purposes and maintain the proper role of the copyright law within the broader framework of our intellectual property system. Yet the law may not have reached an angle of repose. The same interests that sought to expand protection beyond the traditional boundaries of copyright doctrine through litigation are seeking similar expansions of the law through new legislation and other means.

As background for understanding this evolution, this article begins by discussing important differences between markets for traditional products and network products. It then assesses the normative implications for protecting network technologies and why copyright is poorly suited and was never intended to protect network features of computer software. The third section describes the evolution of copyright law for network aspects of computer software, from some early decisions suggesting that copyright law might protect network features of computer software to more recent decisions foreclosing such protection under traditional copyright principles. The concluding section explains why traditional copyright law will play a limited and diminishing role in the appropriability strategies of software companies, while drawing attention to several new clouds looming over the vibrancy of competition and innovation in the computer software industry.

I. The role of network effects in markets for computer software

Among the most important challenges for the courts in grappling with copyright protection for computer software has been in understanding the role of interoperability and compatibility standards in the market for computers and related information tech-

developed by other companies), formed to counter the protectionist agenda of IBM, Digital Equipment Corp. (DEC), Apple Computer Corp., Lotus Development Corporation, and the Computer and Business Equipment Manufacturers Association (CBEMA) and the Software Publishers Association, two leading industry trade association. *See generally* J. BAND & M. KATOH, INTERFACES ON TRIAL: INTELLECTUAL PROPERTY AND INTEROPERABILITY IN THE GLOBAL SOFTWARE INDUSTRY xvii, 120–22 (1995).

nologies, and how traditional copyright doctrines apply to network features of computer software. Markets for products with network effects differ fundamentally from traditional markets because of the potential for demand-side externalities. This section explains the source and economic significance of these differences, the ways in which computer software products can generate network externalities, and the implications of these attributes for competitive strategies in software markets.

A. *Differences between markets for traditional and network products*

In a traditional market, the utility that consumers derive from the purchase and consumption of a product is largely independent of the behavior of other consumers. For example, Billy's purchase and consumption of a vanilla ice cream cone does not depend on the flavor of ice cream cone that Susie purchases. The market accounts for these effects and economic efficiency would not be improved through government intervention.[10] Competition among sellers of ice cream leads to an efficient allocation of resources.

In a market where network effects are significant, such as telecommunications products, consumer choices may have external effects upon other consumers in the market. Suppose that Billy owns a facsimile machine that can receive faxes only from other machines using the ABC communications protocol. In deciding whether to purchase a facsimile machine using the ABC protocol or a competing system using another protocol, Susie will consider the benefits *to her* of being able to communicate (by facsimile)

[10] The price that Billy pays may be affected by the tastes of other consumers if there are economies of scale in the production of particular flavors of ice cream over some ranges of the production function. It is rare, however, that these ranges are larger than the size of the market. Moreover, the price system broadly defined can usually insure efficiency in these circumstances. The effects are more likely pecuniary, which work through the market and only affect the distribution of value, than technological, which affect the economic efficiency of the economy. *See* Liebowitz & Margolis, *Network Externality: An Uncommon Tragedy*, 8 J. ECON. PERSP. 133 (1994).

with the users of the respective systems, the machine's price, and other attributes. She will not, if guided by her own self-interest, consider the benefits to existing users of the systems, such as Billy, of joining one or the other (or multiple) system(s). Therein lies the externality.[11] New entrants to a network consider only their own costs and benefits in making consumption choices and not the external effects upon existing members of the network.

Although network effects have been significant since the beginning of human civilization,[12] they have taken on growing importance and have received significant scholarly, legal, and policy attention with the rise of the information age.[13] Modern markets where network effects are significant include interface specifications for computer operating systems, protocols for Internet communication, networks of automatic teller machines for banking, and broadcast standards for high-definition television.

In general, network effects arise when the value that consumers place on a product depends upon the number of other consumers purchasing that product. This occurs most directly when the product enables consumers to communicate or interact with others who purchase the same product. This includes the classic telephone network, where the value that any consumer places on

[11] In theory, this externality, like all other externalities, could be addressed through private bargaining, *see* Coase, *The Problem of Social Cost*, 3 J.L. & ECON. 1 (1960), but the transaction costs of such a result would likely be prohibitive.

[12] All means of communication, including spoken languages, sign languages, and other means of communicating, have network effects. Other examples of network effects in formal markets are standardized thread patterns for nuts and bolts, electricity currents (110 versus 220), and electrical plugs (grounded or not).

[13] *See* Katz & Shapiro, *Network Externalities, Competition, and Compatibility*, 75 AM. ECON. REV. 424 (1985); Farrell & Saloner, *Standardization, Compatibility, and Innovation*, 16 RAND J. ECON. 70 (1985); Menell, *Tailoring Legal Protection for Computer Software*, 39 STAN. L. REV. 1329 (1987). For a comprehensive survey of the myriad applications of the network externality concept in law and public policy, *see* Lemley & McGowan, *Legal Implications of Network Effects*, 86 CAL. L. REV. 479 (1998).

the product rises with the number of other people whom they can contact through the network. Other examples of such physical networks are the network of interoperable facsimile machines, e-mail networks, modems, and other communication systems where the product in question has value principally because of its ability to interconnect users.

Network effects can also arise where the products do not actually connect or directly communicate with each other, but consumers' valuation of the product depends on the number of other consumers purchasing the same or interoperable products. Operating system software is a prime example of such a "virtual network."[14] Consumers place a higher value on computer systems for which they can expect a greater quantity and quality of application programs to run—now and over their planning horizon. Application programmers can be expected to provide more and better programs for more popular operating systems. Hence, *ceteris paribus*, consumers will favor a computer featuring a more popular operating system. Other examples of virtual networks include application programs that enable users to share data files with other programs and users and/or that are widely adopted so that employees can switch jobs without retraining,[15] ATM cards and automatic teller machines which they can access, and credit cards and the merchants who accept them. The defining feature of virtual networks is that the value of a product depends significantly on the value of the system in which it is a part. A critical determinant of the value of a product, therefore, is the range and value of components with which it is interoperable.[16]

Network markets differ from traditional markets in a number of significant ways. Network markets feature increasing returns as

[14] *See* Katz & Shapiro, *supra* note 13, at 424 (coining the term "virtual network").

[15] *See* Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045, 1070 n.133 (1989); *Cf.* Lemley & O'Brien, *Encouraging Software Reuse*, 49 STAN. L. REV. 255, 260–66, 285–90 (1997).

[16] *See* Katz & Shapiro, *Systems Competition and Network Effects*, 8 J. ECON. PERSP. 93 (1994).

the number of consumers rise, which often leads to dominance by one system. This is especially true for physical networks since the sole or primary purpose of the product is to enable communication. *Ceteris paribus*, consumers will prefer telephone networks or fax machines or modems offering the largest network. A universal network often has the greatest value because it connects the largest number of users. Increasing returns also apply to virtual networks, although particular products may feature other attributes that are independent of the network features. The availability of better application programs to run on an operating system platform will lead more consumers to prefer that operating system, which in turn will spur a greater supply of quality application programs for that operating system. For this reason, one standard or system is likely to predominate in virtual network markets.[17] Because of the large potential returns to becoming the dominant firm in a network market, firms may develop a new standard rather than adopt an existing standard in circumstances in that social welfare would be improved through compatibility.[18] A related feature of network markets is that the high value that consumers place upon standardization may make it particularly difficult for improved products to break into the market. Such

[17] *See* Besen & Farrell, *Choosing How to Compete: Strategies and Tactics in Standardization*, 8 J. ECON. PERSP. 117 (1994).

[18] By adopting a compatible standard, a firm entering the market enlarges the size of a network comprising both its product and its rivals' products. This will have the effect of increasing the desirability of the rivals' products to consumers, thereby reducing the adopter's market share (although of a larger market) relative to what it would have been had the firm adopted an incompatible product standard. Thus even though the net social welfare of adopting a rival's standard may exceed the net social welfare of introducing an incompatible standard, the entrant may nonetheless prefer to adopt the incompatible standard because the entrant cannot appropriate all the benefits of compatibility, some of which accrue to past and present purchasers of rivals' products. *See* Katz & Shapiro, *supra* note 13, at 435. *See also* Katz & Shapiro, *Product Introduction with Network Externalities*, 40 J. INDUS. ECON. 55 (1992); Katz & Shapiro, *Technology Adoption in the Presence of Network Externalities*, 94 J. POL. ECON. 822 (1986).

markets may suffer from excess inertia that can trap the industry in an obsolete or inferior standard.[19]

B. Means by which computer software can generate network externalities

Computers provide a variety of means for creating networks capable of generating positive externalities. As computers have become increasingly versatile, computer software has become a flexible means for creating such network effects. Such effects can be generated at three distinct levels: (1) *protocols* or other technical interface specifications that enable machines to communicate with each other and that allow application programs to run on the specific operating system of a particular computer; (2) *systems*, such as command menus, for structuring the use of an application program; and (3) *computer-human interfaces*, such as screen displays and other input-output devices, that enable the user to operate an application program effectively.

1. PROTOCOLS AND INTERFACE SPECIFICATIONS At the hardware level, networks are formed through the interconnectivity or interoperability of computers and/or computer devices. Examples include the ability of decentralized computers to access data from a common storage device, to exchange data (e.g., through a modem), and to access the central processing unit (CPU) of other computers for purposes of running programs. Such network effects are closest to the network features of a traditional telecommunications network, and the Internet and other network-oriented computer systems are increasingly serving this function.[20] These networks are formed by technical protocols that allow for the interoperability of computers. They can be embedded directly into the hardware or firmware (i.e., hard wiring) of the machine or, as is increasingly the case, in software. Interoperability plays a significant role in determining the boundaries of markets within the computer industry. For example, if a modem is not designed to

[19] *See* Farrell & Saloner, *supra* note 13.

[20] *See* Lemley, *Antitrust and the Internet Standardization Problem*, 28 CONN. L. REV. 1041, 1041–65 (1996).

connect to a particular computer, then users of that computer would have no interest in purchasing that product (unless they were to purchase another computer that functions with that modem).

Technical interoperability is also crucial at the level at which hardware and software interact. Modern computers run on a particular operating system platform, such as Microsoft Windows. An application program can operate on such a computer only if it is designed to communicate with the specific protocols and interface specifications of the operating system. During the formative stage of the microcomputer market in the early 1980s, compatibility between operating systems and application programs was a critical determinant of consumer demand for computers. Consumers placed substantial value on being able to run application programs designed for the IBM personal computer (PC), which featured a proprietary basic instruction operating system (BIOS), a layer of interface specifications embedded in a read-only memory (ROM) chip mediating between the operating system (PC-DOS, IBM's version of Microsoft's Disk Operating System (DOS)) and the central processing unit of the computer. Early clones of the IBM PC failed to fully emulate all aspects of the IBM BIOS, which caused some application programs to malfunction. Clones were not able to penetrate the market significantly until their manufacturers successfully emulated the interface specifications of the BIOS chip. Fear among consumers that their computers would not be able to run application programs designed for the IBM PC enabled IBM to capture a large portion of the microcomputer market in the early- to mid-1980s. This market share eroded after competitors developed interoperable versions of the BIOS chip.

Interoperability between operating systems and application programs continues to play an important role in computer markets today, although the market structure and dynamics have changed since the 1980s.[21] Microcomputers have continued to grow in

[21] *See generally* Baseman, Warren-Boulton, & Woroch, *Microsoft Plays Hardball: The Use of Exclusionary Pricing and Technical Incompatibility to Maintain Monopoly Power in Markets for Operating System Software*, 40 ANTITRUST BULL. 265 (1995); J. BAND & M. KATOH, INTER-

importance and most computers today run on the Microsoft Windows operating system platform. This dominance in the operating system market has provided Microsoft with important competitive advantages in the development and upgrading of application programs and the development of other complementary products, such as Internet software and services.[22] As it improves the Windows operating system platform, Microsoft is able to stay ahead of competitors in developing software that runs on and interact with this platform.

2. SYSTEMS Network externalities can also be generated through the design of application programs. Consumers purchase and use computers in order to communicate, access information, play games, and perform data processing and other tasks more efficiently. The value that consumers place upon computers and software products increases with the ease and effectiveness with which they can perform these functions. Consequently, software engineers design programs so that consumers can learn a system easily and build upon this knowledge as new applications are developed.[23] They often use intuitive conceptual metaphors—such as the desktop as a way of organizing office functions, cutting and pasting as a way of representing the moving of text within a document, or an infinitely manipulable matrix as a metaphor for accounting forms—so as to take advantage and build upon models and behaviors already familiar to or easily learned by their target market of users.[24] The metaphor provides a logical process for organizing tasks in the operation of an application program,[25]

FACES ON TRIAL: INTELLECTUAL PROPERTY AND INTEROPERABILITY IN THE GLOBAL SOFTWARE INDUSTRY (1995).

[22] *See* Baseman, Warren-Boulton, & Woroch, *supra* note 21; Gary L. Reback, *Microsoft's Illegal Conduct: Text of Netscape's Letter to the U.S. Justice Department*, 27 ANTITRUST L. & ECON. REV. 97 (1996).

[23] *See* Menell, *supra* note 15, at 1053–56.

[24] *See* Samuelson, Davis, Kapor, & Reichman, *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2310, 2324–26 (1994).

[25] For example, in designing a program for use in determining the dimensions and spacing of roof supports in the construction trade, the

as is reflected in the menu hierarchies of modern spreadsheet programs.

Over time, users internalize the particular representation of tasks in a computer language or application program, often memorizing the most commonly used series of keystrokes or through the development of macros that customize their most common tasks, which makes them beholden to that way of implementing the metaphor and encourages employers to adopt computer program systems that are widely known by prospective employees in order to reduce training costs.[26] For this reason, it is common for people seeking jobs in data processing, word processing, accounting, and design fields to list on their resumes those computer languages and application programs that they have mastered. Network externalities arise from the enhanced labor mobility and reduced training costs that flow from common, or at least compatible, computer systems across different work environments. They also flow from the increased ability of people in different places to communicate more efficiently through compatible file formats.

3. MACHINE-HUMAN INTERFACES A related way in which software engineers can foster network externalities is through the design of the computer-human interface.[27] The interface between the computer and the user consists of a variety of input/output devices, including a keyboard, pointing tools (such as a mouse, joystick, and interactive pen), disk drives, audio equipment, microphone, and screen displays. For example, the graphical user interface based upon the desktop metaphor—which was first developed by Xerox in the early 1970s and refined in Apple's Macintosh line of products and the Microsoft Windows platform[28]—has proven to be a particularly valuable interface for users.

programmer structured the program to follow the logical steps of a draftsman in developing a layout by hand and presented these tasks in a user-friendly manner. *See* MiTek Holdings, Inc. v. ARCE Engineering Co., 89 F.3d 1548, 1551 n.6 (1996).

[26] *See* Gandal, *Hedonic Price Indexes for Spreadsheets and an Empirical Test for Network Externalities*, 25 RAND J. ECON. 160 (1994).

[27] *See* Menell, *supra* note 15, at 1053–56.

[28] *See* Apple Computer, Inc. v. Microsoft Corp., 799 F. Supp. 1006, 1017–18 (N.D.Cal. 1992).

User interfaces can create network externalities in a variety of ways. User interfaces implement systems for interacting with a computer and operating an application program. They structure the way in which data are input and organized. Thus they establish specifications for using a program and performing tasks. This can be critical for technical compatibility (e.g., file formats) as well as user familiarity with a way of operating a program. In addition, the interface may use logical or externally constrained ways of structuring tasks (e.g., basing the interface for a tax preparation program on the structure of government tax forms or the use of standard terminology from a particular trade, discipline, or specialty). As users learn these tasks, they benefit to the extent that the interface is widely available in other work places and across related programs.[29] In fact, minimizing users' learning time is an important principle of user interface design.[30] Employee mobility is enhanced, the cost of training is reduced, and data files can be more widely used by the wide diffusion of standard computer-human interfaces.

C. Competitive strategies in markets featuring network externalities

Network effects and the potential for network externalities in markets for computer services significantly influence the competitive strategies of computer companies. The relatively short history of the computer industry suggests that network effects have played a dramatic role in shaping market structure and the mar-

[29] *See* Churbuck & Freedman, *Suits Against 1-2-3 Imitators May Have Wide User Impact*, PC WEEK, Jan. 20, 1987, at 1 (noting that " '[t]he real cost of software is not in the package but in the price of training' " (quoting Wayne Maples, information-center consultant at the Federal Reserve Bank in Dallas)); *Interview: Apple Computer, Inc., President and Chairman John Scully—On Fitting into the IBM World of Computing*, PERS. COMPUTING, Apr. 1986, at 145, 147 ("It's becoming apparent that the real cost is not the hardware or even the software. The real cost is teaching the user").

[30] *See* B. SHNEIDERMAN, DESIGNING THE USER INTERFACE: STRATEGIES FOR EFFECTIVE COMPUTER INTERACTION 14–15 (1986); Menell, *supra* note 15, at 1053–54.

gins on which competition occurs. In traditional markets, where network effects are minimal or nonexistent, competition turns principally upon price, quality, and service considerations. In markets in which network effects are significant, competition plays out in other dimensions as well, particularly strategies to establish, maintain, and control standards for the industry.[31]

Consumers in network markets seek to benefit from network effects. They typically want to insure that they will be a part of or have access to the dominant network. They fear being stranded on an obsolete or declining network (or standard), which will deprive them of access to the latest or best interoperable products or impede their mobility in transferring their investments in physical or human capital. Hence, they focus not only on the price and quality of the products that they purchase but also the size of the existing installed base of the network product and what they expect to be the size of the network over their planning horizon. They may even purchase products that they consider to be inferior to currently available alternatives if they believe that the inferior product will enable them to achieve greater streams of network benefits into the future.[32]

[31] *See* Besen & Farrell, *supra* note 17.

[32] *See* Farrell & Saloner, *supra* note 14; Katz & Shapiro, *Technology Adoption in the Presence of Network Externalities*, 94 J. POL. ECON. 822 (1986); Krugman, *History versus Expectations*, 106 Q.J. ECON. 651 (1991). Among the more commonly cited examples of superior product standards that were displaced by inferior rivals offering greater network benefits are Sony's Betamax videocassette recorder (eclipsed by JVC's VHS system which became widely adopted through liberal licensing practices) (*but see* Liebowitz & Margolis, *supra* note 10, at 147–48 (questioning whether Sony's Betamax was a superior standard), Digital Resource Incorporated's Disk Operating System (eclipsed by Microsoft's DOS, which benefited from IBM's adoption for its PC line), and Apple's Macintosh graphical user interface (eclipsed by Microsoft's Windows interface). Even long-time users of the Macintosh interface who continue to believe in its superiority on a stand-alone basis have begun migrating to the increasingly dominant Windows standard in order to derive greater network benefits. *See* Lewis, *Clan Macintosh Feels the Pain*, N.Y. TIMES, Apr. 2, 1998, at G1, col.4 .

Consumers' expectations drive the strategies of competitors in the horizontal market—the market for products establishing a network (e.g., a telecommunications system, a computer operating system)—as well as vertical markets—markets for products that connect to the network (e.g., application programs, peripherals). Unlike traditional markets where a range of products of varying price and quality typically remain competitive over time, network markets often result in a single standard becoming dominant. In some cases, a single firm controls that standard, such as IBM's long dominance in the mainframe computer industry, IBM's relatively brief dominance of the early personal computer market, and Intel's dominance of "x86" microprocessor markets. In others, multiple firms compete in offering products on a particular standard, as occurred with the VHS videocassette recorder. The profit opportunities in network markets can be tremendous given the tendency toward the dominance of a single standard.[33]

Firms typically choose among three classes of strategies in deciding how to compete in horizontal markets: (1) pursuing a strategy of market dominance through establishing a new proprietary standard; (2) adopting an existing standard either through imitation (where it is legally permissible) or licensing; or (3) working with other firms in the industry either informally, contractually, or through formal industry organizations,[34] or through governmental standardization bodies to develop an open or quasi-open standard.[35] Strategies (2) and (3) are not mutually exclusive (and are often pursued in tandem) and create a more traditional market setting in which firms compete over price, quality, and services in an effort to win market share on a common platform. This achieves greater competition on a particular platform and fosters the realization of network externalities, but may dull competition in innovating better platforms.[36] The market dominance strategy is often riskier, but can produce the highest payoff for the prevailing competitor.

[33] *See* Besen & Farrell, *supra* note 17, at 119.

[34] *See* Farrell & Saloner, *Coordination Through Committees and Markets*, 19 RAND J. ECON. 235 (1988).

[35] *See* Besen & Farrell, *supra* note 17.

[36] Katz & Shapiro, *supra* note 16, at 111.

A firm's choice of strategy will depend on a range of factors, including the firm's reputation among consumers for serving the type of network market that it has targeted, its available resources (and access to capital markets) to make the investments in distribution and marketing necessary to persuade consumers that the firm will prevail in the competition over standards,[37] the strength of its technology for establishing a standard (although such technology need not be superior to others on the market), and complementary assets within the firm or strong strategic alliances in vertical markets.[38] The firm's strategy will also depend upon the

[37] Among the strategies firms use to establish their product as the de facto industry standard are massive advertising campaigns, penetration pricing (pricing below cost in order to speed adoptions by consumers) (*see* Farrell & Saloner, *Installed Base and Compatibility: Innovation, Product Preannouncements, and Predation*, 76 AMER. ECON. REV. 940 (1986), including in some cases giving away the product), issuing impressive product preannouncements to entice consumers and discourage competitors (Id.; Warren-Boulton, Baseman & Woroch, *supra* note 21), providing adopters with various forms of insurance (such as short-term leases or pricing arrangements that tie the price of the system to the number of adopters) (Dybvig & Spatt, *Adoption Externalities as Public Goods*, 20 J. PUB. ECON. 231 (1983); Thum, *Network Externalities, Technological Progress, and the Competition of Market Contracts*, 12 INT'L J. INDUS. ORG. 269 (1994)), licensing of the product in order to grow the network more rapidly (and to create competition in the expansion of the network) (Farrell & Gallini, *Second-Sourcing as a Commitment: Monopoly Incentives to Attract Competition*, 103 Q.J. ECON. 673 (1988); Shepard, *Licensing to Enhance Demand for New Technologies*, 18 RAND J. ECON. 360 (1987)), vertical integration and strategic investments into markets for complementary products in order to assure consumers that valuable application programs will be available (Church & Gandal, *Integration, Complementary Products and Variety*, 1 J. ECON. & ECON. MGMT. STRAT. 85 (1992)), and other forms of commitment to prevailing in the competition over which standard will emerge as the de facto standard in the marketplace. *See generally* Church & Ware, *Network Industries, Intellectual Property Rights and Antitrust Policy, in Competition Policy, Intellectual Property Rights and International Economic Integration*, INDUSTRY CANADA (forthcoming 1998).

[38] The firm seeking to establish its standard for a system (hardware) will typically promote development of a wide range of high quality complementary products (software) that interconnect to its system while discouraging the development of such products for its competitor's system.

availability of intellectual property protection, contractual means, and/or technological controls (e.g., encryption technology) for precluding, limiting, or delaying access by competitors to the firm's standard.

Combining its reputation for serving the mainframe market and its technological and marketing capabilities with copyright and trade secrecy protection for its basic instruction operating system (BIOS) chip, IBM successfully pursued the market dominance strategy when it entered the microcomputer market in the early 1980s, enabling it to earn supracompetitive profits. IBM's strategic hold on the industry quickly unraveled when competitors successfully reverse engineered the BIOS chip, making fully-compatible IBM clones available on the market.

II. The proper scope of copyright protection for network features of computer software[39]

The availability of legal protection for those aspects of computer software that provide the basis for a system standard plays a

See Besen & Farrell, *supra* note 17, at 117 & n.1. It may do this through vertical integration, licensing application program developers, or a combination. If the producer of the system is vertically integrated, as Microsoft has become in operating system and application program software, it may seek dominance in both the system and software markets. Alternatively, as has occurred in video game markets, manufacturers of game systems (consoles) may seek to maximize their overall profits through exclusion and price discrimination by regulating or auctioning access to vertical markets (games) through licensing agreements. *See* Warren-Boulton, Baseman, & Woroch, *Economics of Intellectual Property Protection for Software: The Proper Role for Copyright*, 3 STANDARDVIEW 1, 10 (June 1995).

[39] This section draws upon Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045 (1989) and Brief Amicus Curiae of Professors Peter S. Menell and Dennis S. Karjala in Support of Respondent, Lotus Development Corp. v. Borland Int'l, Inc., U.S. Supreme Court, No. 94-2003 (October Term, 1995). *See also* Karjala & Menell, *Applying Fundamental Copyright Principles to Lotus Development Corp. v. Borland International Inc.*, 10 HIGH TECH. L.J. 177, 182–86 (1995).

critical role in computer markets. There is little incentive for firms to pursue noncooperative, market dominance strategies if they cannot maintain the platform that they sponsor as a proprietary standard. Competitors could free ride on the substantial investments required to establish a standard in the marketplace. Although nonintellectual property forms of protection may be available (such as encryption or licensing), formal intellectual property protection offers one of the most secure means of maintaining a proprietary standard. Such protection thus strengthens the ability of firms to develop proprietary standards, which can affect a firm's willingness to participate in efforts to develop industry-wide standards, the realization of network externalities, incentives for research and development of complementary products, and incentives to engage in research and marketing efforts to improve a standard.

Two sets of considerations guide the analysis of the proper scope of copyright protection for network features of computer software.[40] First, drawing upon the constitutional purpose authorizing copyright protection, the purpose of federal intellectual property law is "to Promote the Progress of Science and the useful Arts."[41] Second, recognizing that copyright is but one component of a coherent system for promoting such progress, the scope of copyright must be interpreted in the context of the broader landscape of intellectual property protection.[42] The utilitarian philosophy reflected in the intellectual property clause of the U.S. Constitution focuses attention on the limitations of the market in

[40] The term "network features" is used in this article in an economic (as opposed to purely technological) sense. From this perspective, the term refers to those aspects of a computer program or hardware/software system (such as interface specifications, systems, or computer-human interfaces) that are *capable* of generating network externalities. The critical issue is whether product manufacturers seek to take advantage of potential demand-side externalities in their design, manufacturing, or legal protection strategies.

[41] U.S. CONST. art. I, § 8, cl. 8.

[42] *See generally* R. MERGES, P. MENELL, M. LEMLEY & T. JORDE, INTELLECTUAL PROPERTY IN THE NEW TECHNOLOGICAL AGE (1997) [hereinafter cited as IPNTA].

promoting technological progress. For computer software, two market failures stand out: the traditional public goods problem associated with information goods and the network externalities flowing from many computer, information, and telecommunications technologies. Combining an understanding of the economics of network markets with an appreciation of the manner in which the intellectual property system operates as a whole establishes that copyright law should not protect network features of computer software.

A. *Traditional intellectual property analysis*

Intellectual property protection addresses the appropriability problem inherent in the creation of works embodying valuable information by providing limited periods of exclusive rights to those who create original artistic expression and technological advances. During this period of exclusive rights, the creators have an opportunity to recoup their investments and earn a positive rate of return without fear of copying or imitation by competitors. Despite the similar basic approach of patent and copyright law, however, they seek to promote different types of creativity as reflected in their statutory frameworks. In exchange for protection of claimed functional attributes of an invention, the patent law requires an inventor to demonstrate to a technically trained examiner that the asserted invention is novel, nonobvious, useful, and adequately described to enable one skilled in the art to practice the invention.[43] After meeting these exacting requirements, the patent holder receives exclusive rights to practice the invention for 20 years from the application date. By contrast, the copyright law requires only that a work be "original"[44] in order to receive protection for the expressive elements of the work. Copyright

[43] 35 U.S.C. §§ 101, 102, 103, 112. *See generally* IPNTA, *supra* note 42, at 157–229.

[44] As recently articulated by the Supreme Court, "[o]riginal, as that term is used in copyright, means only that the work was independently created by the author . . . and that it possesses at least some minimal level of creativity." Feist Publications v. Rural Tel. Serv. Co., 499 U.S. 340, 345 (1991).

protection extends for the life of the author plus 50 years (or 75 years in the case of entity authors), but does not extend to "any idea, process, system, method of operation, concept, principle, or discovery."[45] This exclusion from copyright protection of functional attributes of works insures that technological inventions failing to meet the requirements of the patent law do not indirectly receive even longer protection under the copyright law.[46]

Unlike in the domain of literary or artistic creativity where there is relatively little societal interest in minor variations on a single theme or plot,[47] technology advances incrementally through building directly upon the works of others.[48] Secondary inven-

[45] 17 U.S.C. § 102(b).

[46] The Supreme Court articulated this fundamental channeling doctrine in the seminal case of Baker v. Selden, 101 U.S. 99, 104 (1879), in which the holder of a copyright in a book describing a method of accounting sought to prevent others from using his system of accounts:

> . . . Charles Selden, by his books, explained and described a peculiar system of book-keeping, and illustrated his method by means of ruled lines and blank columns, with proper headings on a page, or on successive pages. Now, whilst no one has a right to print or publish his book, or any material part thereof, as a book intended to convey instruction in the art, any person may practise and use the art itself which he has described and illustrated therein. The use of the art is a totally different thing from a publication of the book explaining it. The copyright of a book on book-keeping cannot secure the exclusive right to make, sell, and use account-books prepared upon the plan set forth in such book. Whether the art might or might not have been patented, is a question which is not before us. It was not patented, and is open and free to the use of the public. And, of course, in using the art, the ruled lines and headings of accounts must necessarily be used as incident to it.

[47] *See* Kepner-Tregoe, Inc. v. Carabio, 203 U.S.P.Q. 124, 131 (E.D. Mich. 1979) ("[T]here is no societal interest in many variants on a single theme or plot, nor is there the likelihood that by extending broad protection, entry to the market for literary works will be foreclosed").

[48] *See* Lemley, *The Economics of Improvement in Intellectual Property Law*, 75 TEX. L. REV. 989 (1997); Scotchmer, *Standing on the Shoulders of Giants: Cumulative Research and the Patent Law*, 5 J. ECON. PERSP. 21 (1991); Menell, *supra* note 15, at 1057.

tions—including essential design improvements, refinements, and adaptations to a variety of uses—are often as crucial to the generation of social benefits as the pioneering discovery.[49] The patent system fosters the cumulative invention process by not allowing protection for straightforward or obvious advances,[50] having a shorter duration, and requiring disclosure of how an invention operates. This system is undermined to the extent that technological advances receive protection under copyright law. Because exact coding is required for compatibility, copyright protection for interoperability standards would effectively foreclose competition in network markets.

The network features of computer software serve technological ends and hence are more appropriately protected, if at all, through the patent system. In the case of protocols or interface specifications, software provides the means for machines or hardware and software to communicate. In the case of systems for application programs, software provides a method for accomplishing tasks logically and efficiently and establishes standards for file sharing and other forms of interoperability across application programs. In the case of computer-human interfaces, software enables users to operate programs more effectively. Some aspects of computer-human interfaces are expressive, such as textual material or fanciful characters in a video game, and fall within the scope of traditional copyright protection, but to the extent that they represent a logical system for structuring tasks (such as a desktop metaphor), reflect the principles of user-friendly software design, or become a de facto standard in the industry for accomplishing basic tasks, then their functionality should exclude such aspects of the interface from the scope of copyright protection. To provide protection for such works under copyright would undermine the integrity of the intellectual property system as a whole by effectively affording protection for functional attributes of works under copyright law. Patent protection is unquestionably

[49] *See* Menell, *supra* note 13, at 1338.

[50] *See* Graham v. John Deere Co., 383 U.S. 1 (1966).

available for advances in computer software.[51] Although the examination of software patent claims by the Patent and Trademark Office has been problematic,[52] these considerations only reinforce the case, on traditional intellectual property grounds, for excluding network features of computer software from the scope of copyright protection. Copyright's minimal originality threshold and long duration would result in overbroad protection for network and other functional features of computer software.

B. The implications of network effects for intellectual property protection

Legal protection for network features of computer software can both exacerbate and ameliorate the market distortions created by network effects. While it is desirable for the law to encourage innovation—including innovation in computer system platforms, application program systems, and computer-human interfaces—the law should also promote the adoption of uniform standards so

[51] *See, e.g., In re* Alappat, 33 F.3d 1526 (Fed. Cir. 1994) (*en banc*); Diamond v. Diehr, 450 U.S. 175 (1981); U.S. Patent & Trademark Office, Examination Guidelines for Computer-Related Inventions, 61 Fed. Reg. 7478 (Feb. 28, 1996). *See generally* Sterne, Kessler, Axenfeld, Lee & Sokohl, *Preparing and Prosecuting Electronic and Computer Related Patent Applications: Avoiding and Overcoming Statutory Subject Matter Rejections*, 33 IDEA 297 (1993); Laurie & Siino, *A Bridge Over Troubled Waters? Software Patentability and the PTO's Proposed Guidelines* (Parts 1 & 2), 12 COMPUTER LAW. (Sept. & Oct. 1995).

[52] *See Public Hearing on Use of the Patent System to Protect Software-Related Inventions Before Bruce A. Lehman, Assistant Secretary of Commerce and Commissioner of Patents and Trademarks* (1994) (statement of Bernard Galler).

http://www.uspto.gov/text/pto/hearings/va_ galler.html (noting the lack of an adequate prior art database for evaluating software patents) [hereinafter cited as Galler testimony]; Markoff, *High-Technology Outcry Against the U.S. Patent System*, N.Y. TIMES, Jan. 3, 1994, at C16; *League for Programming Freedom, Against Software Patents*, 14 HASTINGS COMM./ENT. L.J. 297, 299–302, 306–08 (1992) (commenting that the PTO has not adequately recruited or trained examiners of software-related inventions); Kahin, *The Software Patent Crisis*, TECH. REV., Apr. 1990, at 52.

as to foster network externalities. Because of the dynamics of network markets, firms may seek to limit the realization of network externalities and distort the research and development incentives of their competitors and potential competitors in order to claim a larger share of the market for themselves. Thus, it is important that intellectual property law encourage the realization of network externalities while providing appropriate incentives for the development of improved standards and other aspects of network technologies.

1. REALIZATION OF NETWORK EXTERNALITIES In the absence of technological differences between system standards, consumers would prefer one nonproprietary or open platform in a market rather than two (or more). A single platform that is open to competition will generate competition among firms in supplying products. Such competition will lower prices and expand the network, thereby maximizing positive network externalities. By contrast, where multiple platforms exist or if a firm is able to establish a monopoly on the dominant platform, consumers will face higher prices and will not be able to realize the full benefits of network externalities.

Providing copyright protection for network features of computer software would increase the incentives for firms to establish proprietary platforms even where overall efficiency would be improved through a single open standard. Without providing any significant technological advance, a firm sponsoring an "original" system could potentially tilt the market through penetration pricing, investments in vertical markets, and other techniques[53] toward its proprietary standard. Once a proprietary standard becomes dominant, the sponsor will have greater ability to raise prices and leverage its control into other markets. Microsoft's recent efforts to parlay its control over the Windows operating system into control over the Internet browser market and other means of accessing the Internet bear out this concern.[54] In addi-

[53] *See supra* note 37.

[54] *See* Baseman, Warren-Boulton & Woroch, *supra* note 21; Reback, *supra* note 22; Klein & Bansal, *International Antitrust Enforcement in the Computer Industry*, 41 VILL. L. REV. 173 (1996).

tion, the sponsor of a proprietary standard will have little incentive to open up the platform or cooperate with other firms to develop improved standards, all of that would enhance the realization of network externalities.[55]

2. INNOVATION Through control of a widely adopted platform, a firm can exercise substantial control over innovation in that platform and a range of complementary products that connect to or are compatible with that platform. The owner of a proprietary standard can prevent competitors from participating in the market for the platform by refusing to license or by undermining noninfringing products of competitors,[56] which will discourage competition in research to further develop the standard. The adverse effects on innovation may be particularly pronounced because monopolists often have less incentive to develop drastic improvements in their technology because such innovations displace the monopolist's existing profits.[57] The owner of a proprietary standard can also exercise substantial control over innovation in complementary products. It may, for example, engage in predatory product innovation, such as introducing incompatibility codes into a platform that undermine the application programs or peripheral products of rivals in competition with the platform sponsor's products.[58] Furthermore, because of consumers' desire to maintain network benefits into the future, firms controlling a dominant platform can discourage research efforts by rivals and adoption of rivals' products by strategically preannouncing the future availability of new and improved versions of the dominant platform or new products in complementary markets that may benefit from some connection to the

[55] *See* Farrell, *Arguments for Weaker Intellectual Property Protection in Network Industries*, 3 STANDVIEW 46 (1995); Farrell, *Standardization and Intellectual Property*, 30 JURIMETRICS J. 35 (1989); Menell, *supra* note 13, at 1359–63.

[56] *See* Baseman, Warren-Boulton & Woroch, *supra* note 21, at 277–78.

[57] *See* JEAN TIROLE, THE THEORY OF INDUSTRIAL ORGANIZATION (1988).

[58] *See, e.g.*, Brock, *Dominant Firm Response to Competitive Challenge: Peripheral Equipment Manufacturers' Suits Against IBM*, *in* THE ANTITRUST REVOLUTION (J. Kwoka Jr.& L. White eds., 1989).

dominant platform.[59] Even if these products do not appear by the expected date or fail to live up to promises, competitors suffer because many consumers will delay purchases in order to await the platform sponsor's products.[60]

A recent antitrust case concerning the microprocessor industry illustrates well the myriad ways in which control over an industry standard can affect innovation and competition in horizontal and vertical markets. Judge Nelson of the federal district court in Alabama made the following factual findings regarding Intel's exercise of market power in the microprocessor market:

> Intel has purposely changed its CPU architecture by using proprietary sockets and otherwise, converting from the previously "open architecture" to a new "closed architecture." This "closed architecture," for practical purposes, allows Intel, by exercising its intellectual property rights in its "closed architecture," to wield absolute power over who will and who will not be allowed to participate in that part of the high-end computer industry that is based upon the "x86" microprocessor. Inasmuch as it requires two or more years and millions of dollars to design and develop a mother board and graphics subsystem to accept and take advantage of a CPU such as the Pentium II or any possible alternative, 18 OEMs [original equipment manufacturers], such as Intergraph, who rely entirely on Intel for their supply of microprocessors and chip sets have become technologically and financially locked in to the Intel CPU, its associated chip sets, and the P6 Bus, and they have no feasible alternative to it. . . .
>
> OEMs, such as Intergraph, are capable of competing in the marketplace only if they also have access to: (1) Intel's advance confidential technical information, which is necessary to develop new products and to service existing products; (2) advance samples of Intel's development chips, which are needed by Intergraph to develop its own next-generation products using those chips . . . ; and (3) early releases of

[59] *See* Farrell & Saloner, *Installed Base, Product Preannouncements, and Predation*, 76 AM. ECON. REV. 940 (1986); Baseman, Warren-Boulton & Woroch, *supra* note 21, at 272; Prentice, *Vaporware: Imaginary High-Tech Products and Real Antitrust Liability in a Post-Chicago World*, 57 OHIO ST. L.J. 1163 (1996).

[60] Although product preannouncements need not be anticompetitive and can serve important functions in educating consumers, they can be potent tools in manipulating the innovation and marketing decisions of rivals in network markets.

> Intel chips, related products, and related technical information, which is necessary for Intergraph to test and produce its products . . . .
>
> Intel provides its products, as well as technical and design information, to Intergraph under NDAs [nondisclosure agreements], which are terminable at will by Intel. These NDAs are documents drafted by Intel and presented to Intergraph and other customers on a take-it-or-leave-it basis. Intel simply dictates the terms under which it provides its products and design information to customers such as Intergraph.
>
> The evidence suggests strongly that Intel has used the threatened or actual termination of NDAs as a contractual weapon, coercing customers such as Intergraph to accede to Intel's demands and restraining competition. The court takes judicial notice of the fact that, when Intergraph and Digital Equipment separately asserted their patent rights against Intel, Intel immediately used the termination provisions of their respective NDAs to deny both Intergraph and Digital further technical information, samples, and products and demanded the return of all Intel confidential information provided to them. . . .
>
> In view of Intel's previous policy of providing much of the same type of information now subject to NDAs in a much less restricted manner and in view of the fact that Intel has offered no reasonable explanation of any present need for the use of the NDAs, it seems reasonable to conclude that Intel's present use of one sided and terminable-at-will NDAs and its retaliatory cancellation of the NDAs are unreasonable and anticompetitive contractual restraints using Intel's monopoly in CPUs and related design and technical information. Furthermore, the chilling effect which Intel's arbitrary enforcement of the NDAs in this manner must have on other members of the industry, who are dependent upon Intel for microprocessors, is obvious. . . .
>
> The court also finds that Intel has attempted to leverage its monopoly power in the "x86" CPU market to prevent Intergraph from competing in the graphics subsystem and workstation markets and to control and dominate competition in these markets through discriminatory and favored agreements and understandings with some of Intergraph's competitors. This reduces competition in the markets in which Intergraph competes, depriving customers of alternative and improved technology in these markets, stifling innovation, reducing competition in price and quality, and impairing competition generally. . . .[61]

[61] Intergraph v. Intel Corp., U.S.D.C. N.D.Ala. CV 97-N-3023-NE (April 10, 1998) (citations and footnotes omitted). After finding that Intel's chip architecture and technical information are "essential facilities" in the microprocessor market, the court held that "Intel's refusal to supply advanced CPUs and essential technical information to Intergraph

Given these threats to innovation, the threshold for providing intellectual property protection for network technologies should be suitably high, and certainly well above the copyright standard of originality.[62] In view of the cumulative nature of technological advancement,[63] intellectual property law must be careful not to choke off secondary innovations by according excessive or unjustified protections for first-generation inventions. This risk is particularly great in the computer software field, where many successful products build upon intuitive (and hence often obvious) metaphors and readily recognizable (and hence often obvious) symbols in developing functional, user-friendly programs and interfaces.

3. ADOPTION OF IMPROVED STANDARDS The very forces that drive network markets to a single standard and thereby foster the realization of network externalities can also "trap" the industry in an obsolete or inferior standard.[64] The installed base built upon the "old" standard—reflected in past purchases of application programs embodying the old standard and human capital investments (worker training)[65] specific to that standard—can create excess inertia that makes it much more difficult for any one producer to break away from the old standard by introducing a new incompatible interface. This inertia can result even if the new interface offers significant technological advancements.[66] Sometimes firms

likely violates § 2 of the Sherman Act, because they are not available from alternative sources and cannot be feasibly duplicated, and because competitors cannot effectively compete in the relevant markets without access to them." *Id.*

[62] *See* Farrell & Katz, *The Effects of Antitrust and Intellectual Property Law on Compatibility and Innovation*, in this issue of *The Antitrust Bulletin*; Farrell, *supra* note 55; Menell, *supra* note 13, at 1359–67; Menell, *supra* note 15, at 1088.

[63] *See* Merges, *Intellectual Property Rights and Bargaining Breakdown*, 62 TENN. L. REV. 75 (1994); Merges & Nelson, *On the Complex Economics of Patent Scope*, 90 COLUM. L. REV. 839, 868–906 (1990); Scotchmer, *supra* note 48.

[64] *See* Farrell & Saloner, *supra* note 13.

[65] *See supra* note 29.

[66] *See* DAVID HEMENWAY, INDUSTRYWIDE VOLUNTARY PRODUCT STANDARDS 30, 39 (1975); Farrell & Saloner, *supra* note 37, at 940; Farrell &

can create bridges to a new standard,[67] such as by enabling files to be read by the new system, but such retrospective interoperability may come at a cost in terms of the functionality of the new platform. Thus, network markets can retard innovation and slow or prevent adoption of improved platforms.

The availability of carefully tailored legal protection for product standards can alleviate the excess inertia effect by assuring innovators of improved standards a limited monopoly if their products succeed. Without the availability of legal protection, innovators' profits would be competed away quickly as other firms free ride on the innovator's investments and introduce their own versions of the platform or standard. Such protection, however, should only be granted for significant technological advances so as not to undermine the realization of network externalities. Copyright law, with its minimal threshold for protection, is poorly suited to providing such protection. The more exacting patent requirements of novelty and nonobviousness better insure that excessive market power will not be created.[68]

Saloner, *supra* note 13, at 71–72, 75–79. As this literature points out, it is not necessarily efficient for all technological improvements in standards to be adopted immediately because of the cost of replacing the existing stock of products based upon the older standard. Thus, if the new platform offers only minor advances but would make the large stock of installed capacity obsolete, then it is better for society to stick with the older standard until a significant improvement is available or to replace existing stock as it wears out. The desirability of a change in platforms depends significantly on the transition path.

[67] *Cf.* Farrell & Shapiro, *Standard Setting in High Definition Television*, 8 Brookings Papers on Econ. Activity 1 (1992); Farrell & Saloner, *Converters, Compatibility, and Control of Interfaces*, 40 J. Indus. Econ. 9 (1992).

[68] There may be advantages to allowing access to even a proprietary standard after it has become widely adopted and sufficient profits have been appropriated so as to foster realization of network externalities. *See* Menell, *supra* note 15, at 1096. In any case, the patent systems higher threshold for protection and shorter duration are better suited to the protection of network features of computer software than copyright law's originality standard and long duration.

C. Implications for copyright protection of network features

From the perspective of the traditional public goods justification for intellectual property or considerations of network externalities, network features of computer software should not receive direct protection under copyright. Copyright law plays a specialized role within the larger framework of intellectual property protection and is not designed or intended to protect technological innovations. Although not tailored to address the particular concerns raised by network technologies, patent law is better suited to insuring that protection for network features of computer software is not improvidently granted. Protocols, interface specifications, and systems for structuring application programs should not be protectable under copyright law. Furthermore, copyright law should not stand in the way of competitors seeking to develop interoperable computer systems and products. Competitors should be permitted to reverse engineer computer programs in order to determine how to make interoperable products.[69]

The protection of audiovisual elements of computer-human interfaces is somewhat more complicated. These works fall within the traditional domain of copyright law and should continue to be protectable as audiovisual works to the extent that they are sufficiently original and not excluded by section 102(b) of the Copyright Act (copyright shall not extend to "any idea, process, system, method of operation, concept, principle, or discovery")

[69] *See* Graham & Zerbe, *Economically Efficient Treatment of Computer Software: Reverse Engineering, Protection, and Disclosure*, 22 RUTGERS COMP. & TECH. L.J. 61 (1996); BAND & KATOH, *supra* note 9, at 167–226; Warren-Boulton, Baseman & Woroch, *Economics of Intellectual Property Protection for Software: The Proper Role for Copyright*, 3 STANDARDVIEW 1, 5–6 (Jun. 1995); Cohen, *Reverse Engineering and the Rise of Electronic Vigilantism: Intellection Property Implications of "Lock-Out" Technologies*, 68 S. CAL. L. REV. 1091 (1995); Karjala, *Copyright Protection of Computer Documents, Reverse Engineering, and Professor Miller*, 19 U. DAYTON L. REV. 975 (1994); Teter, *Merger and the Machines: An Analysis of the Pro-Compatibility Trend in Computer Software Copyright Cases*, 45 STAN. L. REV. 1061 (1993); McManis, *Intellectual Property Protection and Reverse Engineering of Computer Programs in the United States and the European Community*, 8 HIGH TECH. L.J. 25 (1993); Menell, *supra* note 15, at 1371.

and related doctrines of merger, scenes à faire, and *Baker v. Selden*.[70] Principles of human factor analysis and software engineering should not be protectable by copyright law.[71] The more difficult question arises when a largely expressive (i.e., nonfunctional) compilation of elements at the time of creation becomes a de facto standard through widespread learning by users. Standardization of such features then becomes an important consideration in entering the market with a competing or complementary product. Copyright protection should at that point in time give way so as to foster the realization of network externalities.[72] Thus, the analysis of whether idea and expression have merged (or copying is a fair use) should be based on market realities at the time of competition, not at the time that the original interface is created.[73]

Notwithstanding these limitations on the scope of copyright protection, copyright law will continue to play a useful indirect role in enhancing the appropriability of investments in software research and development. Protection against literal copying of noncopyrightable elements of program code provides software manufacturers with increased lead time given the time that it takes competitors to reverse engineer an interoperability standard and develop noninfringing competing or attaching products.[74] Going beyond this level of protection under copyright law, however, would risk extending monopoly power well beyond what is needed to encourage innovation, undermine the realization of network externalities, and impede innovation in complementary markets.

[70] *See* Consensus Statement, *supra* note 7, at 26–32.

[71] *See* Menell, *supra* note 15, at 1053–57.

[72] *Id.* at 1101–02.

[73] *See* Warren-Boulton, Baseman & Woroch, *supra* note 38, at 5; Teter, *supra* note 69, at 1070; Menell, *supra* note 15, at 1101–02.

[74] *See* Johnson-Laird, *Software Reverse Engineering in the Real World*, 19 U. DAYTON L. REV. 843 (1994).

III. The evolution of copyright protection for network features of computer software

For much of the early history of the computer industry, copyright did not play an important role in the development of the technology.[75] The earliest computers were dedicated calculating machines that were hardwired to perform one or a few specific tasks. They did not utilize separate computer programs. Even after programmable computers became increasingly available in the 1950s, they were large and expensive machines maintained by their manufacturers. The manufacturers typically provided software and service on a contractual basis and hence there was little need for formal intellectual property protection to appropriate an adequate return from this type of investment. Trade secret protection was a viable means of protecting intellectual work embodied in software because most computer programs were customized and sold to particular customers through detailed licensing agreements.

With the proliferation of mini- and microcomputers in the 1970s and the rise of a separate, mass-market software sector, the need for formal means of protecting the intellectual work embodied in computer software developed. Patent protection was largely foreclosed by the resistance of the Patent and Trademark Office to software-related inventions, the Supreme Court's 1972 decision in *Gottschalk v. Benson*[76] excluding pure algorithm claims from the subject matter of patent law, and the relatively high cost and delay in obtaining patent protection.

Faced with the difficult challenge of fitting computer and other new information technologies under the existing umbrella of intellectual property protection, Congress in 1974 established the National Commission on New Technological Uses of Copyrighted Works (CONTU) to study the implications of the new technologies and recommend revisions to the federal intellectual property

[75] *See generally* P. MENELL, R. MERGES & M. LEMLEY, LEGAL PROTECTION FOR COMPUTER TECHNOLOGY (forthcoming 1999); BAND & KATOH, *supra* note 9, at 18–25.

[76] 409 U.S. 63 (1972).

laws. After conducting hearings and receiving expert reports, a majority of the panel of copyright authorities and interest group representatives concluded that "computer programs, to the extent that they embody an author's original creation, are proper subject matter of copyright."[77] CONTU was clear, however, that the fundamental limitation reflected in the idea/expression dichotomy that copyright law cannot protect "any idea, procedure, process, system, method of operation, concept, principle, or discovery"[78] shall apply with equal force with regard to computer programs.[79]

Congress implemented CONTU's recommendations in 1980 by adding a definition of "computer program" to section 101 of the Copyright Act and amending section 117 of the Act to authorize the owner of a copy of a computer program to make another copy or adaptation of the program for the purpose of running the program on a computer.[80] Congress defined "computer program" as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result."

The copying of literal program code and other aspects of computer programs, such as structural features of a program, operating elements (e.g., menu command systems), program outputs (e.g.,

[77] NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS, FINAL REPORT 1 (1979) [hereinafter cited as CONTU REPORT].

[78] 17 U.S.C. § 102(b).

[79] *See* CONTU REPORT, *supra* note 77, at 20. "[C]opyright protection for programs does not threaten to block the use of ideas or program language previously developed by others when that use is necessary to achieve a certain result. When other language *is* available, programmers are free to read copyrighted programs and use the ideas embodied in them in preparing their own works." *Id.* (emphasis in original). "One is always free to make the machine do the same thing as it would if it had the copyrighted work placed in it, but only by one's own creative effort rather than by piracy." *Id.* at 21.

[80] H.R. REP. No. 1307, 96th Cong., 2d Sess. 23, *reprinted in* 1980 U.S. Code Copy. & Admin. News 6460, 6482 (noting that the 1980 amendments to the Copyright Act were intended to implement CONTU's recommendations).

screen displays), and user interfaces, quickly became the focus of numerous lawsuits and the courts were called upon to determine the scope of copyright protection for computer software. The first generation of cases involved literal copying of program code and the courts did not have much difficulty in finding that wholesale reproduction of a computer program, in whatever form the computer program was embodied,[81] was a violation of the Copyright Act. The language in some of these cases, however, was uncritically expansive,[82] and some later courts[83] failed to apply doctrines limiting copyright protection for functional works in a manner that was faithful to section 102(b) of the Copyright Act and its jurisprudential antecedents and the definition of "computer program" in the Act. Over the past 7 years, the courts have largely overcome these early missteps and interpreted the copyright law so as not to protect network features under copyright law. This section will examine how the law has evolved with respect to the principal network aspects of software.

A. *Protocols and interface specifications*

Copyright protection for the protocols or computer code that allows for machine-to-machine, machine-to-software, or software-to-software interoperability has been an active area of litigation. Although some early cases suggested that copyright protection for such elements of a computer program was expansive and a basis for excluding competitors, these cases have been effectively supplanted by an influential and growing body of more recent cases

[81] NEC v. Intel Corp., 654 F. Supp. 590 (N.D. Cal. 1989) (microcode); Apple Computer v. Franklin Computer Corp., 714 F.2d 1240, 1246–48 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984) (program stored in ROM); Williams Electronics, Inc. v. Arctic Int'l, Inc., 685 F.2d 870 (3d Cir. 1982) (object code, program stored in ROM).

[82] *See, e.g.*, Apple Computer v. Franklin Computer Corp., 714 F.2d 1240, 1246–48 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984).

[83] *See* Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 1222 (3d Cir. 1986); Lotus Development Corp. v. Paperback Software Int'l, 740 F. Supp. 37 (1990); Lotus Development Corp. v. Borland Int'l, 799 F. Supp. 203 (1992).

that exclude the elements of interoperability from the scope of copyright. The cases fall into two general categories: those governing the protectability under copyright law of technical interface specifications, and those governing whether competitors may decompile or reverse engineer a copyrighted program in order to determine interface specifications.

1. PROTECTABILITY OF INTERFACE SPECIFICATIONS In the first major test of copyright protection for computer software, Franklin Computer Corporation copied nearly verbatim fourteen computer programs developed by Apple Computer Corporation for its Apple II line of products. Franklin sought to make its computer "compatible" with the Apple II, for which a large supply of independently developed application programs were available. Apple sued for copyright infringement. Franklin defended principally on the ground that operating system programs, as opposed to application programs, are not within the proper domain of copyright law. The lower court and the Third Circuit ruled in Apple's favor.[84] The defendant made no attempt to determine which elements of the program were protectable and which were not. Nonetheless, in addressing the issue of whether achieving interoperability would justify some limited copying, the court commented that:

> The idea which may merge with the expression, thus making the copyright unavailable, is the idea which is the subject of the expression. The idea of one of the operating system programs is, for example, how to translate source code into object code. If other methods of expressing that idea are not foreclosed as a practical matter, then there is no merger. *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.*[85]

Since two entirely different programs may achieve the same "certain result[s]" (e.g., generate the same set of protocols needed for

[84] Apple Computer v. Franklin Computer Corp., 545 F. Supp. 812 (E.D. Pa. 1982), *aff'd*, 714 F.2d 1240 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033 (1984).

[85] 714 F.2d at 1253 (emphasis added).

interoperability), the court was not justified in making such an expansive and uncritical statement about the scope of copyright protection for computer programs. CONTU was clear that "[o]ne is always free to make the machine do the same thing as it would if it had the copyrighted work placed in it, but only by one's own creative effort rather than by piracy."[86] In addition, applying the merger analysis at such a high level of abstraction (where the idea of the program is how to translate source code into object code) would essentially block the development of interoperable systems, creating a powerful property right through copyright protection.

A few years later the Third Circuit reinforced this misguided application of the merger doctrine in assessing copyright protection for application programs. In *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*,[87] the owner of a dental laboratory hired a custom software firm to develop a computer program that would organize the bookkeeping and administrative tasks of its business. Whelan, the principal programmer, interviewed employees about the operation of the laboratory and then developed a program to run on the laboratory's IBM Series One computer. Under the terms of an agreement, Whelan retained the copyright in the program and agreed to use its best efforts to improve the program while Jaslow Laboratory agreed to use its best efforts to market the program. Rand Jaslow, an officer and shareholder of the laboratory, set out to create a version of the program that would run on other computer systems. Whelan sued for copyright infringement. At trial, the evidence showed that the Jaslow program did not literally copy Whelan's code, but there were overall structural similarities between the two programs. As a means of distinguishing protectable expression from unprotectable idea, the court reasoned:

> *[T]he purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea.* Where there are many means of achieving the desired purpose, then the particular means cho-

[86] *See* CONTU REPORT, *supra* note 77, at 21.

[87] 797 F.2d 1222 (3d Cir. 1986), *cert. denied,* 479 U.S. 1031 (1987).

sen is not necessary to the purpose; hence, there is expression, not idea.[88]

In applying this rule, the court defined the idea as "the efficient management of a dental laboratory," for which countless ways of expressing of the idea would be possible.[89] Drawing the idea/expression dichotomy at such a high level of abstraction implies an expansive scope of copyright protection. Furthermore, the court's conflation of merger analysis and the idea/expression dichotomy implicitly allows the protection under copyright of procedures, processes, systems, and methods of operation, that are expressly excluded under section 102(b). Although the case did not directly address copyright protection for computer code establishing interoperability protocols for computer systems, the court's mode of analysis dramatically expanded the scope of copyright protection for computer programs. If everything below the general purpose of the program was protectable under copyright, then it would follow that particular protocols were protectable because there would be other ways of serving the general purpose of the program. Such a result would effectively bar competitors from developing interoperable programs and computer systems.

The *Whelan* test was roundly criticized by commentators[90] and other courts began developing alternative approaches to the

[88] *Id.* at 1236 (emphasis in original; citations omitted).

[89] *Id.*

[90] *See, e.g.*, Englund, *Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 MICH. L. REV. 866, 881 (1990); Menell, *supra* note 15, at 1074, 1084–85; Kretschmer, *Copyright Protection for Software Architecture: Just Say No!*, 1988 COLUM. BUS. L. REV. 823, 837–39; Spivack, *Does Form Follow Function? The Idea/Expression Dichotomy in Copyright Protection of Computer Software*, 35 UCLA. L. REV. 723, 747–55 (1988); Gage, *Whelan Associates v. Jaslow Dental Laboratories: Copyright Protection for Computer Software Structure—What's the Purpose?*, 1987 WIS. L. REV. 859, 860–61 (1987); 3 NIMMER ON COPYRIGHT § 13.03(F), at 13–62.34. The principal defenders of the Whelan approach were lawyers for the software companies seeking broad protection under copyright law. *See* Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since*

scope of copyright protection that better comported with the fundamental principles of copyright protection. A few months after the *Whelan* decision, the Fifth Circuit confronted a similar claim of copyright infringement based upon structural similarities between two programs designed to provide cotton growers with information regarding cotton prices and availability, accounting services, and a means for conducting cotton transactions electronically.[91] In declining to follow the *Whelan* approach, the court found that the similarities in the programs were dictated largely by standard practices in the cotton market (what the court called "externalities"), such as the "cotton recap sheet" for summarizing basic transaction information, which constitute unprotectable ideas.[92]

CONTU?, 106 HARV. L. REV. 977 (1993) (counsel for Lotus Development Corp.); Brown, *"Analytical Dissection" of Copyrighted Computer Software—Complicating the Simple and Confounding the Complex*, 25 ARIZ. ST. L.J. 801 (1993) (counsel for Apple Computer Corp.); Clapes, Lynch & Steinberg, *supra* note 5, at 1546–52 (lawyers for IBM).

[91] Plains Cotton Cooperative Assoc. v. Goodpasture Computer Service, Inc., 807 F.2d 1256 (5th Cir. 1987).

[92] *Id.* at 1262. The court found persuasive the decision in Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1003 (N.D.Tex. 1978), in which Judge Higginbotham analogized the "input formats" of a computer program (the organization and configuration of information to be inputted into a computer) to the "figure-H" pattern of an automobile stick shift.

> Several different patterns may be imagined, some more convenient for the driver or easier to manufacture than others, but all representing possible configurations. . . . The pattern (analogous to the computer "format") may be expressed in several different ways: by a prose description in a driver's manual, through a diagram, photograph, or driver training film, or otherwise. Each of these expressions may presumably be protected through copyright. But the copyright protects copying of the particular expressions of the patterns, and does not prohibit another manufacturer from marketing a car using the same pattern. Use of the same pattern might be socially desirable, as it would reduce the retraining of drivers.

Id. at 1013.

Five years later, the Second Circuit in *Computer Associates International v. Altai, Inc.*[93] expressly rejected the *Whelan* approach to determining the scope of copyright protection for computer programs. Computer Associates, a leading developer of mainframe software, had developed a program that could operate on different IBM mainframe computers (with different operating systems). With access to the Computer Associates' program, Altai developed a competing program serving a similar purpose, which also operated on multiple IBM mainframes. Computer Associates sued for infringement. The district court criticized Whelan's "simplistic test" for determining similarity between computer programs,[94] rejecting the notion that there is but one idea per program and that as long as there were alternative ways of expressing that one idea, then any particular version was protectable under copyright law. Focusing on the various levels of the computer programs at issue, the court determined that the similarities between the programs were dictated by external factors—such as the interface specifications of the IBM operating system and the demands of functionality—and hence no protected code was infringed.[95]

On appeal, the Second Circuit fleshed out a detailed analytical framework for determining copyright infringement of computer program code:

> In ascertaining substantial similarity under this approach, a court would first break down the allegedly infringed program into its constituent structural parts. Then, by examining each of these parts for such things as incorporated ideas, expression that is necessarily incidental to those ideas, and elements that are taken from the public domain, a court would then be able to sift out all non-protectable material. Left with a kernel, or perhaps kernels, of creative expression after following this process of elimination, the court's last step would be to compare this material with the structure of an allegedly infringing program.[96]

[93] 982 F.2d 693 (2d Cir. 1992).

[94] 775 F. Supp. 544, 558 (E.D.N.Y. 1991).

[95] *Id.* at 561–62.

[96] 982 F.2d at 706.

The court's abstraction-filtration-comparison test recognized that an idea could exist at multiple levels of a computer program and not solely at the most abstract level and that the ultimate comparison is not between the programs as a whole but must focus solely on whether protectable elements of the program were copied. Of most importance with regard to fostering interoperability, the court held that copyright protection did not extend to those program elements where the programmer's "freedom to choose" is

> circumscribed by extrinsic considerations such as (1) mechanical specifications of the computer on that a particular program is intended to run; (2) compatibility requirements of other programs with which a program is designed to operate in conjunction; (3) computer manufacturers' design standards; (4) demands of the industry being serviced; and (5) widely accepted programming practices within the computer industry.[97]

Directly rejecting the dictum in *Apple v. Franklin*,[98] the Second Circuit recognized that external factors such as interface specifications, de facto industry standards, and accepted programming practices are not protectable under copyright law. The formulation of the Second Circuit test judges these external factors at the time of the allegedly infringing activities (i.e., *ex post*), not at the time that the first program is written.[99]

Commentators warmly embraced the *Altai* decision[100] and the abstraction-filtration-comparison approach has been universally

[97] *Id.* at 709–10.

[98] *See supra* note 1.

[99] The court emphasized that the first to write a program for a particular application should not be able to " 'lock up' basic programming techniques as implemented in programs to perform particular tasks." 982 F.2d at 712 (quoting Menell, *Analysis of the Scope of Copyright Protection for Application Programs*, 41 STAN. L. REV. 1045, 1087 (1989)).

[100] *See* Bender, *Computer Associates v. Altai: Rationality Prevails*, THE COMPUTER LAWYER, Aug. 1992, at 1; Menell, *The Challenges of Reforming Intellectual Property Protection for Computer Software*, 94 COLUM. L. REV. 2644, 2652 (1994); Lemley, *Convergence in the Law of Software Copyright?*, 10 HIGH TECH. L.J. 1 (1995). The principal exceptions are lawyers for software companies that have advocated for a broad scope of copyright protection for their client's products. *See* Clapes &

adopted by the courts since 1992.[101] Although a few courts have misapplied the test in specific instances, the *Altai* test has supplanted the overbroad merger analysis set forth in *Whelan.* In the context of network technologies, this doctrinal shift has effectively excluded protocols from the scope of copyright protection. In *Gates Rubber v. Bando Chem. Indus., Ltd.*,[102] the Tenth Circuit expressly adopted the *Altai* approach and expanded the range of external factors to be used in filtering out unprotectable elements to include hardware standards and mechanical specifications, software standards and compatibility requirements, industry programming practices, and practices and demands of the industry being serviced.[103] The court also noted that processes used in designing a computer system, or components therein (e.g., modules, algorithms), must also be filtered out as unprotectable under section 102(b).[104] While not ruling that interface specifications are uncopyrightable as a matter of law, the Eleventh Circuit's decision in *Bateman v. Mnemonics, Inc.*[105] joined other circuits following *Altai* in holding that "external considerations such as compatibility may negate a finding of infringement."[106] The court

Daniels, *Revenge of the Luddites: A Closer Look at Computer Associates v. Altai*, The Computer Lawyer, Nov. 1992, at 11; Brown, *supra* note 90, at 816–18; Miller, *supra* note 90.

[101] More than a dozen courts have expressly followed the *Altai* approach. *See, e.g.,* Baystate Technologies, Inc. v. Bentley Systems, Inc., 947 F. Supp. 1079 (D.Mass. 1996); Lemley, *supra* note 100 (surveying cases).

[102] 9 F.3d 823 (10th Cir. 1993).

[103] 9 F.3d at 836–43. *See also* Mitel v. Iqtel, 124 F.3d 1366, 1375 (10th Cir. 1997).

[104] 9 F.3d at 836–37.

[105] 79 F.3d 1532 (11th Cir. 1996).

[106] *Id.* at 1547. In the accompanying footnote, the court commented:

> Note that we use the word "may." Such a finding will depend on the particular facts of a case, and thus it would be unwise for us to try to formulate a bright-line rule to address this issue, given the importance of the factual nuances of each case. In no case, however, should copyright protection be extended to functional

commented that "[i]t is particularly important to exclude methods of operation and processes from the scope of copyright in computer programs because much of the content of computer programs is patentable. Were we to permit an author to claim copyright protection for those elements of the work that should be the province of patent law, we would be undermining the competitive principles that are fundamental to the patent system."[107]

2. PERMISSIBILITY OF REVERSE ENGINEERING A related issue bearing on the extent to which copyright protection may impede the development of interoperable computer systems concerns the extent to which competing manufacturers are able to reverse engineer a computer system to determine the codes governing interoperability. Most computer software is distributed in object code form only, that is not directly readable by humans. If a software manufacturer is able to prevent competitors from learning the interface specifications necessary for interoperability because of more general restrictions on the copying of program code containing the protocols, then the fact that the protocols are not protectable under copyright law would be nugatory since competitors would be precluded from learning the interoperability protocols. In some contexts, a computer program can be understood through input/output testing or other means (for example, physically and chemically peeling the layers of a chip and studying the design of the chip with a microscope) that do not require the making of copies of the computer code in which the protocols are embedded. In most circumstances, however, the only feasible means of deciphering the protocols governing interoperability is disassembly of the program, which involves translating the machine-readable binary object code into a human readable form.[108] If the making of such copies (or translations) is an infringement, then the protocols would be effectively protected by copyright law.

results obtained when program instructions are executed and such results are processes of the type better left to patent and trade secret protection.

[107] *Id.* at 1541 n.21.

[108] *See generally* Johnson-Laird, *supra* note 74.

In the 1980 amendments to the Copyright Act implementing the CONTU recommendations, Congress authorized the owner of a copy of a computer program to make another copy or adaptation of the program for the purpose of running the program on a computer.[109] In *Vault Corp. v. Quaid Software Ltd.*,[110] Vault, the manufacturer of a computer program designed to prevent unauthorized duplication of another program on the same diskette, alleged that Quaid had infringed its copyright in the copy protection program by loading it into its computer's memory for the purpose of reverse engineering the copy-protection device so as to circumvent it. Vault argued that section 117 did not authorize such copying of the program because it was not for the "intended purpose" of running the program.[111] The court declined to construe section 117 so narrowly on the ground that the statutory language did not contain any such restriction.[112]

Beyond the authorization to make copies of computer software as a means for utilizing the program within a computer (loading the program into the internal memory of the computer), courts have afforded competitors substantial leeway to make copies and translations of object code for the purpose of studying how they operate and to develop interoperable products.[113] In *Atari Games*

[109] Section 117. See text accompanying note 80.

[110] 847 F.2d 255 (5th Cir. 1988).

[111] *Id.* at 261.

[112] *Id.* Where the company making the intermediate copies is not an "owner," then the authorization to make internal copies does not apply. *See* MAI Systems Corp. v. Peak Computer Inc., 991 F.2d 511 (9th Cir. 1993), *cert. dismissed*, 114 S. Ct. 671 (1994). It remains an open question whether licensees may properly invoke § 117. This distinction is increasingly significant in view of the widening industry practice of vending software through shrink-wrap licenses. *See* BAND & KATOH, *supra* note 9, at 170–71, Lemley, *Intellectual Property and Shrinkwrap Licenses*, 68 S. CAL. L. REV. 1239 (1995).

[113] *See* Cohen, *supra* note 69; Hayes, *The Legality of Disassembly of Computer Programs*, 12 COMPUTER L.J. 1 (Oct. 1993); Consensus Statement, *supra* note 7, at 24–25. Section 117 does not apply to the making of copies outside of a computer. *But see* McManis, *Intellectual Property*

v. Nintendo,[114] Nintendo protected access to its video game console through proprietary interface specifications embedded in a computer program. After Atari Games deciphered the interface specifications and developed different (and noninfringing) computer code that enabled its games to run on Nintendo's game console, Nintendo sued for copyright infringement on the ground that Atari Games copied protected elements of the Nintendo computer code in the process of decompilation. In assessing the appropriateness of making intermediate copies for the purpose of decompiling a competitor's computer program, the Court of Appeals for the Federal Circuit emphasized the principle that the fair use doctrine generally "permits an individual in rightful possession of a copy of a work to undertake necessary efforts to understand the work's ideas, processes, and methods of operation."[115] The court noted that "[a]n author cannot acquire patent-like protection by putting an idea, process, or method of operation in an unintelligible format and asserting copyright infringement against those who try to understand that idea, process, or method of operation."[116] Applying these principles, the court reasoned that "[w]hen the nature of a work requires intermediate copying to understand the

Protection and Reverse Engineering of Computer Programs in the United States and the European Community, 8 HIGH TECH. L.J. 25, 94–95 (1993) (suggesting that § 117 might allow decompilation).

[114] 975 F.2d 832 (Fed. Cir. 1992).

[115] 975 F.2d at 842. *See* Consensus Statement, *supra* note 7, at 23–25.

[116] 975 F.2d at 842. The consensus statement of leading intellectual property professors participating in the LaST Frontier Conference on Copyright Protection of Computer Software commented that:

> [w]ith regard to traditional copyrightable works, such as works of visual art and textbooks, it is unnecessary to make any copies in order to extract unprotected elements, because these elements can be directly perceived by human beings. With computer programs, however, the process of extracting ideas and other unprotected elements will require the making of one or a limited number of literal copies. The reason is that computer programs cannot be "read" without disassembling or decompilation, a process that may also involve the making a copy of some or all of the code.

Consensus Statement, *supra* note 7, at 24.

ideas and processes in a copyrighted work, that nature supports a fair use for intermediate copying. Thus, reverse engineering object code to discern the unprotectable ideas in a computer program is a fair use."[117] The court placed the following limits on reverse engineering of object code: (1) "Any reproduction of protectable expression must be *strictly necessary* to ascertain the bounds of protected information within the work";[118] (2) Reverse engineering does not authorize commercial exploitation of "protected expression";[119] (3) "To invoke the fair use exception, an individual must possess *an authorized copy* of a literary work."[120]

A short time later, the Ninth Circuit adopted a similar interpretation of the fair use defense in *Sega Enterprises Ltd. v. Accolade, Inc.*[121] As in *Atari Games*, the maker of a video game console (Sega) sought to prevent unauthorized game manufacturers from developing games that could operate on their hardware. In the process of deciphering the interface specifications for the Sega system, Accolade made intermediate copies of the Sega software. Even though the final product did not infringe any protectable computer code, Sega sued on the ground that the intermediate copies infringed Sega's copyright in the console system software. On the basis of a thorough fair use analysis, the Ninth Circuit held that such intermediate copies were excused. Of particular note with regard to the network aspects of the technology, the court emphasized the strong public policy reasons for allowing a competitor to create interoperable works.[122] "If disassembly

[117] 975 F.2d at 843.

[118] *Id.* (emphasis added).

[119] *Id.* at 844.

[120] *Id.* at 843 (emphasis added). Since Atari Games had acquired a copy of Nintendo's source code under false pretenses—by misrepresenting to the Copyright Office that Atari Games was defending a copyright infringement action and that it would use the source code only in putting on its defense—the court refused to allow Atari Games use of the equitable defense of fair use. *Id.* at 841.

[121] 977 F.2d 1510 (9th Cir. 1993).

[122] *Id.* at 1526. Thus, the court rejected the dictum in *Apple v. Franklin*, *supra* note 1, stating that achieving compatibility "is a com-

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."[123] The court concluded that "an attempt to monopolize the market [through copyright] by making it impossible for others to compete runs counter to the statutory purpose of promoting creative expression and cannot constitute a strong equitable basis for resisting the invocation of the fair use doctrine."[124]

Thus, the courts have determined that competitors may reverse engineer computer programs to understand the manner in which they operate and to determine interface specifications so to be able to develop interoperable programs.[125] Where necessary, such reverse engineering may properly involve the creation of intermediate copies of protected computer program code. Decompilation, however, can be laborious, time consuming, and expensive.[126] In addition, prudent developers of interoperable products will typically want to use "clean room" procedures, which add additional time and cost to the development process, in order to avoid the copying of protected com-

mercial and competitive objective which does not enter into the somewhat metaphysical issue of whether particular ideas and expressions have merged."

[123] 977 F.2d at 1526. *See also* Brief Amicus Curiae of Copyright Law Professors, Sega v. Accolade, 977 F.2d 1510 (9th Cir. 1992) (No. 92-15655), *reprinted in* 33 JURIMETRICS J. 147 (Fall 1992).

[124] 977 F.2d at 1524. *See* Cohen, *supra* note 69; Consensus Statement, *supra* note 7; CONTU REPORT, *supra* note 77, at 20 ("[C]opyright protection for programs does not threaten to block the use of ideas or program language previously developed by others when that use is necessary to achieve a certain result. When other language is available, programmers are free to read copyrighted programs and use the ideas embodied in them in preparing their own works").

[125] *See, e.g.,* Bateman v. Mnemonics, Inc., 79 F.3d 1532, 1540 (11th Cir. 1996) (following *Sega v. Accolade*); Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693, 700, 714–15 (2d Cir. 1992); NEC Corp. v. Intel Corp., 10 U.S.P.Q.2d 117 (N.D.Cal. 1989); E.F. Johnson Co. v. Uniden Corp. of America, 623 F. Supp. 1485, 1501 n.17 (D.C. Minn. 1985).

[126] *See* Johnson-Laird, *supra* note 74.

puter code in their own products.[127] Nonetheless, as properly (and currently) applied by the courts, copyright law does not stand in the way of achieving interoperability at the level of hardware-hardware, hardware-software, or software-software interface specifications.

B. Systems

As noted earlier, Congress distinguished in its definition of "computer program" in section 101 of the Copyright Act between the "set of statements or instructions to be used directly or indirectly in a computer" and the "certain result[s]" that they bring about. Thus, the language of the statute indicates that it is the program code itself that was brought within the scope of the Copyright Act in the 1980 amendments and that the behavior of the program[128] (the "certain result[s]")—such as the screen displays and menu command structures—are not covered by the copyright in the program.[129] These behaviors of the program are copyrightable, if at all, because they separately meet the requirements of the Copyright Act.[130]

The *Altai* court appreciated this distinction.[131] Although some courts have not clearly distinguished between copyright protec-

[127] A clean room procedure involves using two sets of computer engineers—one to decompile the target program to determine the interface specifications and a second team that does not have access to the target program, which develops the interoperable program solely on the basis of the interface specifications—to insure that the final product does not contain any infringing code (and that the development team can prove that they independently developed their code). Copyright lawyers have developed detailed procedures for insuring the integrity of this process. *See* Davis, *Scope of Protection of Computer-Based Works: Reverse Engineering, Clean Rooms and Decompilation*, 370 PLI/Pat 115, 151 (1993).

[128] *See* Samuelson et al., *supra* note 24, at 2316–26.

[129] *See* Karjala & Menell, *supra* note 39.

[130] To find otherwise would make little sense since different programs can produce the same behavior, as in interface specifications and screen displays.

[131] [W]e note that our decision here does not control infringement actions regarding categorically distinct works, such as certain

tion for the computer code and the "certain results" that they generate, they have nonetheless applied a sensible reading of section 102(b) of the Act to limit protection of command systems governing the operation of a computer program.

The issue of the copyrightability of command systems for computer software arose most directly in litigation surrounding spreadsheet technology. Building upon the success of the Visicalc program developed for the Apple II computer, the Lotus Corporation marketed an enhanced and faster operating spreadsheet program incorporating many of Visicalc's features and commands into its 1-2-3 program for the IBM PC platform.[132] Lotus 1-2-3 quickly became the market leader for spreadsheets running on IBM and IBM-compatible machines, and knowledge of the program became a valuable employment skill in the accounting and management fields. The 1-2-3 command hierarchy was particularly attractive because it provided a logical structuring of more than 200 commands and it enabled users to developed customized programs (called macros) to automate particular accounting and business planning functions in their workplace. Businesses and users increasingly became "locked-in" to the 1-2-3 command structure as their human capital investments in learning the system and library of macros grew.[133] By the late 1980s, software developers seeking to enter the spreadsheet market could not ignore the large premiums that many consumers placed on being able to use their investments in the 1-2-3 system in a new spreadsheet environment, even where a new spreadsheet product offered significant technical improvements over the Lotus spreadsheet.[134]

> types of screen displays. These items represent products of computer programs, rather than the programs themselves, and fall under the copyright rubric of audiovisual works.

982 F.2d at 703.

[132] *See* Menell, *supra* note 15, at 1057; BAND & KATOH, *supra* note 9, at 155.

[133] *See* Gandal, *supra* note 26.

[134] *See, e.g.,* Hogan, *Product Outlook: Fresh from the Spreadsheet Oven*, PC WORLD, Feb. 1988, at 100–02; Magid, *"Surpass" Spreadsheet Program Lives Up to Name, Beats Lotus 1-2-3*, WASH. POST, Apr. 25, 1988, at 26.

In the mid 1980s, Paperback Software International introduced a spreadsheet program (VP-Planner) that largely emulated the operation of the Lotus 1-2-3 product.[135] Paperback was careful to insure that the program code did not copy the 1-2-3 code. Nonetheless, Lotus sued Paperback for copyright infringement, alleging that VP-Planner inappropriately copied the 1-2-3 menu structure, which included the choice of command terms, the structure and order of those terms, their presentation on the screen, and the long prompts. Relying upon the Third Circuit's merger test in *Whelan* and hence focusing simply upon whether such elements could be expressed in a variety of ways, Judge Keeton of the District Court of Massachusetts found for Lotus. Facing bankruptcy, Paperback agreed not to appeal the judgment as part of a settlement.[136]

After 3 years of intensive development efforts, Borland International, developer of several successful software products including Turbo Pascal and Sidekick, introduced Quattro Pro, its entry into the spreadsheet market. Unlike Paperback's VP-Planner spreadsheet, which offered little beyond the 1-2-3 product, Quattro Pro made substantial design and operational improvements and earned accolades in the computer product review magazines.[137]

[135] *See* Licklider, *Ten Years of Rows and Columns*, BYTE, Dec. 1989, at 324.

[136] *See* Ould, *Legal Dispute Kept Paperback from Lotus Appeal*, PC WEEK, Jan. 21, 1991, at 138.

[137] *See Spreadsheet; Borland International Inc.'s Quattro Pro for Windows and Quattro Pro 4.0 for DOS*, PC-COMPUTING, Dec. 1992, at 140 ("No doubt about it: Quattro Pro for DOS is the best DOS spreadsheet there is. Period."); *Borland's Quattro Pro Tops 2.5 Million Units Shipped*, BUSINESS WIRE, Jul. 1, 1992 ("Since its introduction in October 1989, Quattro Pro has won an unprecedented 42 industry awards and honors worldwide from users and product reviewers. Borland's Quattro Pro continues to outscore competing versions of Lotus 1-2-3 in key testing lab reviews. In two separate reviews, InfoWorld awarded Quattro Pro a spreadsheet report card score of 7.3 (InfoWorld, April 6, 1992), while Lotus 1-2-3 Release 2.4 received a 6.2 (InfoWorld, June 1, 1992). Quattro Pro outscored Lotus 1-2-3 for DOS by significant margins in an independent study conducted by Usability Sciences Corp. Representative spreadsheet users determined Quattro Pro 4.0 to be easier to use, richer in

Also unlike VP-Planner, Quattro Pro offered a new interface for its users, which many purchasers of spreadsheets preferred over the 1-2-3 interface. Nonetheless, because of the large number of users already familiar with the 1-2-3 command structure and those who had made substantial investments in developing macros to run on the 1-2-3 platform, Borland considered it essential to offer an operational mode based on the 1-2-3 command structure as well as macro compatibility.[138] Unlike VP-Planner, Borland's visual representation of the 1-2-3 command mode substantially differed from the 1-2-3 screen displays.

In order to clarify the legal status of its product, Borland brought a declaratory judgment action in California. Through astute jurisdictional maneuvering, Lotus was able to have the case consolidated with the *Paperback* case before Judge Keeton. After protracted litigation,[139] Judge Keeton found for Lotus using a somewhat refined version of the *Whelan* merger test to find that a menu command structure is protectable if there are many such structures theoretically available. He also found that Borland was not permitted to achieve macro compatibility with the 1-2-3 prod-

features, more productive and the preferred spreadsheet over Lotus 1-2-3 for DOS."); *Software Review, Quattro Pro 4.0; Borland International Inc.'s Spreadsheet Software,* COMPUTER SHOPPER, Jun. 1992, at 536 ("Quattro Pro 4.0 simply shames other DOS-based spreadsheets, especially Lotus 1-2-3 r2."); Hogan, *supra* note 134.

[138] *See Software Review: Revamped Quattro Pro Closes in on Lotus 1-2-3*, PC-COMPUTING, Nov. 1989, at 50 (favorable review noting that "Quattro Pro's compatibility with Lotus 1-2-3 Release 2.01 is as good as Lotus' own Release 3—if not better. You can read or write 1A, 2.01, or 2.2 files, use a Lotus-compatible menu, and run 1-2-3 macros without conversion. . . . If you choose to avoid Windows, then Quattro is the leader in spreadsheet publishing and database integration. Its high degree of Lotus compatibility means that *1-2-3 retraining is minimal,* and its moderate hardware requirements (512K of RAM and a hard disk) give it maximum flexibility.") (emphasis added).

[139] *See* Lotus Dev. Corp. v. Borland Int'l, Inc., 788 F. Supp. 78 (D. Mass. 1992); Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F. Supp. 203 (D. Mass. 1992); Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F. Supp. 202 (D. Mass. 1993); Lotus Dev. Corp. v. Borland Int'l, Inc., 831 F. Supp. 223 (D. Mass. 1993).

uct, distinguishing the treatment of external constraints noted in the *Altai* decision on the ground that such constraints had to exist at the time that the first program was created—both the Altai and Computer Associates programs were designed to provide interoperability across IBM platforms. Thus, Judge Keeton effectively ruled that constraints governing the design of computer systems must be analyzed *ex ante* (based on technical considerations at the time the first program is written) and not *ex post* (after the market has operated to establish a de facto standard).

Borland appealed the judgment to the First Circuit.[140] By this point in time, the Second Circuit's *Altai* decision had received a favorable reception in the professional and academic journals and its approach had been adopted by a number of courts. The Ninth Circuit and the Federal Circuit had issued the *Sega* and *Atari Games* decisions, further emphasizing the legitimacy of developing interoperable systems. In addition, the Supreme Court's decision in *Feist Publications, Inc. v. Rural Telephone Service Co.*,[141] denying copyright protection for alphabetically organized telephone directories for lack of originality, repudiated the "sweat of the brow" doctrine[142] and reaffirmed that the "long recognized" principle "that the fact/expression dichotomy limits severely the scope of protection in fact-based works."[143] In addition, the *Borland* case had attracted tremendous interest among academics and interest groups. The court received numerous amicus briefs reflecting a range of interests and perspectives.[144]

[140] 49 F.3d 807 (1st Cir. 1995).

[141] 499 U.S. 340 (1991).

[142] A few lower courts had found that copyright could be established on the basis of substantial effort in gathering facts. *See, e.g.*, Leon v. Pacific Telephone & Telegraph Co., 91 F.2d 484 (9th Cir. 1937); Jeweler's Circular Publishing Co. v. Keystone Publishing Co., 281 F. 83 (2d Cir. 1922). The Supreme Court in *Feist* rejected this "sweat of the brow" theory and held that originality is a requirement of copyright and therefore, unless a factual work exhibits originality as a compilation, it cannot receive protection under the Copyright Act.

[143] 499 U.S. at 350.

[144] Amicus briefs were filed on behalf of computer scientists, intellectual property professors, the Computer Software Industry Association,

The First Circuit viewed the case as one of first impression: "[w]hether a computer menu command hierarchy constitutes copyrightable subject matter."[145] The court properly distinguished *Altai* as dealing with protection of programming code and not the results of such code. Instead, the court saw the subject matter of this case as a "method of operation" falling directly within the exclusions from copyright set forth in section 102(b).[146]

> We think that "method of operation," as that term is used in § 102(b), refers to the means which a person operates something, whether it be a car, a food processor, or a computer. Thus a text describing how to operate something would not extend copyright protection to the method of operation itself; other people would be free to employ that method and to describe it in their own words. Similarly, if a new method of operation is used rather than described, other people would still be free to employ or describe that method.
>
> We hold that the Lotus menu command hierarchy is an uncopyrightable "method of operation." The Lotus menu command hierarchy provides the means by which users control and operate Lotus 1-2-3. If users wish to copy material, for example, they use the "Copy" command. If users wish to print material, they use the "Print" command. Users must use the command terms to tell the computer what to do. Without the menu command hierarchy, users would not be able to access and control, or indeed make use of, Lotus 1-2-3's functional capabilities.
>
> The Lotus menu command hierarchy does not merely explain and present Lotus 1-2-3's functional capabilities to the user; it also serves as the method by which the program is operated and controlled. . . .[147]

The U.S. Supreme Court granted certiorari and affirmed without opinion by an equally divided vote.[148]

a coalition of users' groups, the Software Entrepreneurs' Forum, the American Committee for Interoperable Systems, two coalitions of major computer and software manufacturers, and the Computer and Business Equipment Manufacturers Association.

[145] 49 F.3d at 813.

[146] The court noted that it did not need to determine whether the menu command hierarchy was also unprotectable under copyright law because it was a system, process, or procedure. 49 F.3d at 814.

[147] 49 F.3d at 815.

[148] 516 U.S. 233 (1996) (Justice Stevens did not participate in the consideration of the case).

Subsequent appellate decisions have reached similar outcomes, although they have not fully subscribed to the First Circuit's reasoning. In *MiTek Holdings, Inc. v. ARCE Engineering Co.*,[149] the holder of a copyright in an application program that designed and arranged wood trusses for the framing of building roofs brought an infringement action against the maker of a competing program that featured a similar menu command tree and user interface. Affirming the lower court's decision, the Eleventh Circuit held that the menu and submenu command structure of the truss design program was uncopyrightable under section 102(b) of the Copyright Act because it represents a process.[150] The court did not need to reach the broader question, addressed in *Lotus*, of whether all menu command structures are uncopyrightable as a matter of law. In *Mitel, Inc. v. Iqtel, Inc.*,[151] Mitel, the maker of a widely adopted computer system for automating the selection of telephone long-distance carrier and remotely activating optional telecommunications features such as speed dialing, sued a competing firm which used the identical command codes for copyright infringement. Because Mitel's system had become a de facto standard in the marketplace, Iqtel defended its use of compatible controller codes on the ground that "technicians who install call controllers would be unwilling to learn Iqtel's new set of instructions in addition to the Mitel command code set, and the technician's employers would be unwilling to bear the cost of additional

[149] 89 F.3d 1548 (11th Cir. 1996).

[150] 89 F.3d at 1556–57. The court further noted that the lower court's decision could be sustained on the grounds that the menu and submenu command structures were unoriginal ("The look of the ACES program is basically industry standard computer-aided-design (CAD).") and that idea and expression had merged ("the ACES programs 'mimic the steps a draftsman would follow in designing a roof truss plan by hand' [quoting the conclusion of the district court] . . . . The logical design sequence is akin to a mathematical formula that may be expressed in only a limited number of ways; to grant copyright protection to the first person to devise the formula effectively would remove that mathematical fact from the public domain.").

[151] 124 F.3d 1366 (10th Cir. 1997).

training."[152] As Borland had done, Iqtel's product included both its own set of command codes as well as a "Mitel Translation Mode." While commenting that a method of operation may in some circumstances contain copyrightable expression, the Tenth Circuit nonetheless concluded that the Mitel command codes, which were arbitrarily assigned, lacked the minimal degree of creativity to qualify for copyright protection.[153] The court further held that Mitel's command codes should be denied copyright protection under the *scenes à faire* doctrine because they are largely dictated by external factors such as compatibility requirements and industry practices.[154]

C. Machine-human interfaces

The interface between the computer and the user consists of a variety of input/ouput devices, including a keyboard, pointing tools (such as a mouse, joystick, and interactive pen), disk drives, audio equipment, microphone, and screen displays. Copyright law excludes from protection such obviously functional works as keyboards, pointing objects, speaker systems, and other hardware devices. The courts have also found that data input formats, such as the order and size of data fields, are not protectable under copyright law.[155] The visual images and text of screen displays may qualify as audiovisual, graphic, or literary works under copyright.[156] Some early courts afforded substantial protection to elements of a user interface.[157] Such works remain, however, subject

[152] *Id.* at 1369.

[153] *Id.* at 1373–74.

[154] *Id.* at 1374–76.

[155] *See, e.g.,* Synercom Technology, Inc. v. University Computing Co., 462 F. Supp. 1003 (N.D.Tex. 1978); *cf.* Mitel, Inc. v. Iqtel, Inc., 124 F.3d 1366 (10th Cir. 1997).

[156] *See, e.g.,* Stern Electronics, Inc. v. Kaufman, 669 F.2d 852 (2d Cir. 1982).

[157] *See, e.g.,* Broderbund Software, Inc. v. Unison World, 648 F. Supp. 1127, 1134 (N.D.Cal. 1986) (holding that the choice of typeface on user screen display and choice of works "Choose a Font" as the title for a screen

to the originality requirement and functionality limitations of section 102(b), merger doctrine, and *Baker v. Selden*. As a result of the network features of computer-human interfaces, many aspects of these works are not protectable under copyright law.

The Ninth Circuit addressed some of the limitations on copyright protection for audiovisual displays of computer programs in *Data East USA, Inc. v. Epyx, Inc.*,[158] in which the manufacturer of a video game depicting a karate match sought to prevent another firm from marketing a competing game featuring many of the same audio and visual elements. Notwithstanding the many similarities between the two works, the court held that no infringement had occurred because the similarities flowed from "constraints inherent in the sport of karate itself" and "various constraints inherent in the use of [the particular type of] computer."[159] After filtering out the unprotectable ideas in the work, the court applied a standard of "virtual identity" in determining that the competing work did not infringe.[160]

The most significant case to address the scope of copyright protection for network features of a computer-human interface is *Apple Computer, Inc. v. Microsoft Corp*,[161] in which Apple Com-

for producing cards, brochures, and other printing projects were examples of audiovisual displays "dictated primarily by artistic and aesthetic consideration, and not by utilitarian or mechanical ones"); Digital Communications Associates, Inc. v. Softklone Distributing Corp., 659 F. Supp. 449 (N.D.Ga. 1987) (finding that the arrangement of status screens and commands for a data communication program are protectable expression).

[158] 862 F.2d 204 (9th Cir. 1988).

[159] *Id.* at 209. *See also* Interactive Network v. NTN Communications, 875 F. Supp 1398 (N.D.Cal. 1995), *aff'd*, 57 F.3d 1083 (Fed. Cir. 1996) (finding that football video game was not infringed because similarities between works were based on the rules of football and the idea of an interactive prediction game).

[160] Prior case law in the Ninth Circuit held that "[w]hen idea and expression coincide, there will be protection against nothing other than identical copying." Sid & Marty Krofft Television Productions v. McDonald's Corp., 562 F.2d 1157, 1168 (9th Cir. 1977).

[161] 799 F. Supp. 1006 (1992), *aff'd in part, rev'd in part*, 35 F.2d 1435 (1994), *cert. denied*, 513 U.S. 1184 (1995).

puter alleged that Microsoft's Windows operating system and Hewlett-Packard's NewWave operating system infringed Apple's copyrights in the desktop graphical user interface for its Macintosh computer system. The copyright issue was somewhat muddied by the existence of a licensing agreement authorizing the defendants to use aspects of Apple's graphical user interface. The court determined, however, that the licensing agreement was not a complete defense to the copyright claims[162] and therefore undertook an analysis of the scope of copyright protection for a large range of audiovisual elements of computer screen displays.

In framing the analysis, the district court expressly recognized the relevance of network externalities and the cumulative nature of innovation to the scope of copyright protection:

> Copyright's purpose is to overcome the public goods externality resulting from the non-excludability of copier/free riders who do not pay the costs of creation. Peter S. Menell, An Analysis of the Scope of Copyright Protection for Application Programs, 41 Stan. L. Rev. 1045, 1059 (1989). But overly inclusive copyright protection can produce its own negative effects by inhibiting the adoption of compatible standards (and reducing so-called "network externalities"). Such standards in a graphical user interface would enlarge the market for computers by making it easier to learn how to use them. Id. at 1067-70. Striking the balance between these considerations, especially in a new and rapidly changing medium such as computer screen displays, represents a most ambitious enterprise. Cf. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F.Supp. 37 (D. Mass. 1990).
>
> While the Macintosh interface may be the fruit of considerable effort by its designers, its success is the result of a host of factors, including the decision to use the Motorola 68000 microprocessor, the tactical decision to require uniform application interfaces, and the Macintosh's notable advertising. And even were Apple to isolate that part of its interface's success owing to its design efforts, lengthy and concerted effort alone "does not always result in inherently protectible [sic] expression." [Quoting *Computer Associates v. Altai*, 982 F.2d at 711.]

[162] Apple Computer, Inc. v. Microsoft Corp., 709 F. Supp. 925, 930 (N.D.Cal. 1989); Apple Computer, Inc. v. Microsoft Corp., 717 F. Supp. 1428 (N.D.Cal. 1989); Apple Computer, Inc. v. Microsoft Corp., 759 F. Supp. 1444 (N.D.Cal. 1991).

> By virtue of having been the first commercially successful programmer to put these generalized features together, Apple had several years of market dominance in graphical user interfaces until Microsoft introduced Windows 3.0, the first DOS-based windowing program to begin to rival the graphical capability of the Macintosh. . . . To accept Apple's "desktop metaphor"/"look and feel" arguments would allow it to sweep within its proprietary embrace not only Windows and NewWave but, at its option, also other desktop graphical user interfaces which employ the standardized features of such interfaces, and to do this without subjecting Apple's claims of copyright to the scrutiny which courts have historically employed. Apple's copyrights would hold for programs in existence now or in the future—for decades. One need not profess to know for sure where should lie the line between expression and idea, between protection and competition to sense with confidence that this would afford too much protection and yield too little competition.
>
> The importance of such competition, and thus improvements or extensions of past expressions, should not be minimized. The Ninth Circuit has long shown concern about the uneasy balance which copyright seeks to strike:
>
>> What is basically at stake is the extent of the copyright owner's monopoly—from how large an area of activity did Congress intend to allow the copyright owner to exclude others? *Herbert Rosenthal Jewelry Corp. v. Kalpakian,* 446 F.2d 738, 742 (9th Cir. 1971).
>
> The court declines Apple's invitation to use the advent of the microcomputer and its interface to abandon traditional standards which govern copyrights . . . .[163]

The court proceeded to determine those elements of the graphical user interface which were not protected on the grounds that they were not original or not protectable under section 102(b), the doctrine of *scenes à faire,*[164] or the merger of idea and expression, or due to the limited number of ways in which an idea could be expressed or the external constraints imposed by the computer

[163] 799 F. Supp. at 1025–26.

[164] Under the doctrine of *scenes à faire*, copyright protection is denied to expressions that are "as a practical matter, indispensable or at least standard in the treatment of a given [idea]." Atari, Inc. v. North American Phillips Consumer Electronics Corp., 672 F.2d 607, 616 (7th Cir. 1982), *cert. denied,* 459 U.S. 880 (1982).

system. The court found that all of the alleged similarities between Apple's works and Microsoft's Windows not authorized by the licensing agreement were either not protectable or subject to at least one of the limiting doctrines. As a result, the court applied the "virtual identity" standard in comparing the works as a whole and determined that no infringement had occurred. On appeal, the Ninth Circuit affirmed the district court's dissection of the work in question to determine which elements are protectable, its filtering out of unprotectable elements, and its application of the "virtual identity" standard in this context.[165]

The *Apple* litigation established that many elements of the desktop-based graphical user interface are in the public domain and that the originality requirement and functionality doctrines of copyright law substantially limit the protection afforded the desktop user interface. The Eleventh Circuit has since joined the Ninth Circuit in adopting the "virtual identity standard" for claims of software infringement in a computer-user interface based on a compilation of uncopyrightable elements.[166] While even the virtual identity standard might prevent the realization of network externalities in the context of some interface protocols—where interoperability requires the generation of identical protocols—such a standard does not threaten the realization of network externalities in the context of computer-human interfaces. Minor variations will not unduly inhibit the portability of consumer learning of an interface, while modest protection (i.e., against virtual identity) will provide some incentive for engaging in research to improve the design of computer-human interfaces.

IV. The future of legal protection for network features of computer software

After an inauspicious start, the federal courts have rediscovered and properly applied fundamental limitations on the scope of

[165] 35 F.3d 1435 (9th Cir. 1994), *cert. denied,* 513 U.S. 1184 (1995).

[166] MiTek Holdings, Inc. v. ARCE Engineering Co., 89 F.3d 1548, 1558 (11th Cir. 1996).

copyright protection with regard to aspects of computer software that generate network externalities. As reflected in many of the more recent cases, the courts have finally appreciated the complex relationship among the modes of intellectual property protection and copyright's distinctive role within this larger system. Moreover, the more influential courts have met the challenges posed by network technologies through subtle application of traditional copyright doctrines and principles.

Most of the industrialized nations have largely reached comparable results. The European Community's (EC) Software Directive,[167] adopted in May 1991, excludes interface specifications from copyright protection,[168] permits reverse engineering other than decompilation,[169] and allows decompilation to achieve compatibility for purposes of interoperability with competing as well as attaching programs.[170] Courts in England[171] and Canada[172] have endorsed the *Altai* mode of analysis. Japanese law, although less explicitly than U.S. or EC law, also appears to allow competitors to reverse engineer works for purposes of achieving interoperability.[173]

[167] Council Directive 91/250 of 14 May 1991 on the Legal Protection of Computer Programs, 1991 O.J. (L 122).

[168] *Id.* article 1(2).

[169] *Id.* article 5(3).

[170] *Id.* article 6. Some scholars have criticized the EC Directive's limitation of decompilation right to interoperability on the ground that neither patent nor copyright law have restricted users' privilege to study protected works for any purpose. *See* Huet & Ginsburg, *Computer Programs in Europe: A Comparative Analysis of the 1991 EC Directive*, 30 COLUM. J. TRANSNAT'L L. 327 (1992).

[171] John Richardson Computers Ltd. v. Flanders and Chemtec Ltd., 1993 FSR 497 (English High Court of Justice, Chancery Division); *but see* IBCOS Computers Ltd. v. Barclays Highland Finance Ltd., 1994 FSR 275 (rejecting *Altai* seemingly in conflict with the EC Software Directive).

[172] Deltrina Corp. v. Triolet Systems, Inc., 9 B.L.R.2d 140 (Ont. Ct. of Justice 1993); Matrox Electronic Systems Ltd. v. Gaudreau, 1993 R.J.Q. 2449 (Montreal Court of Justice).

[173] *See generally* BAND & KATOH, *supra* note at 9, at ch. 6.

As a purely doctrinal matter, it may be a bit premature to write copyright's epitaph with regard to protection of network features of computer software. Although the predominant wave of court decisions, supported by persuasive scholarship, has appropriately cabined the role of copyright protection, the Supreme Court has yet to rule decisively on the scope of copyright protection for computer software. Nonetheless, as a practical matter, there are good reasons to expect copyright protection to recede in importance in the protection of network features of computer software. Practitioners increasingly acknowledge the more limited scope of copyright protection[174] and firms have accordingly shifted their competitive strategies accordingly.

This is not to say that traditional copyright protection for computer software will no longer play a significant role in the appropriability strategy of software companies. Copyright provides a low-cost and effective means of preventing piracy of computer program code. In addition, it affords substantial lead time to creators of many programs generating network externalities because of the time and cost required for competitors to reverse engineer interface specifications embedded within complex computer programs.[175]

The CONTU Report stated as a fundamental principle that "[c]opyright should not grant more economic power than is necessary to achieve the incentive to create."[176] The report also stated that "when specific instructions, even though previously copyrighted, are the only and essential means of accomplishing a given task, their later use by another will not amount to infringement."[177] In order to effectuate these principles, copyright protec-

[174] *See, e.g.,* Tanenbaum, *The Challenge of Cyberspace Intellectual Property*, COMPUTER LAW., Feb. 1998, at 14–17; Martone, *Intellectual Property Protection for Computer Software—As Copyright Protection Narrows, Can Patents Fill the Gap?*, Practising Law Institute, Patents, Copyrights, Trademarks, and Literary Property Course Handbook Setric, PLI No. G4-4006 (May 5, 1997).

[175] *See* Johnson-Laird, *supra* note 74.

[176] CONTU REPORT, *supra* note 77, at 12.

[177] *Id.* at 20.

tion should not extend to those aspects of computer programs that generate network externalities. The federal courts have largely reached this conclusion, although they have not yet divined a completely consistent jurisprudential basis.[178] The cases make clear, however, that copyright protection does not extend to the network features of computer programs. As a result, we can expect to see traditional copyright declining in importance in the protection of network technologies, with copyright protection for software limited to the nonfunctional, including nonnetwork, elements of programming code and program outputs such as computer-user interfaces.

The reaffirmation of traditional limitations on the scope of traditional copyright law does not, however, insure that competition and innovation will remain vigorous in network industries. Developments in the legislative sphere, contract law, and the administration of patent law pose serious threats to the functioning of network markets.

The very interests that sought to expand the scope of copyright through litigation have through lobbying efforts had extraordinary influence in defining the United States position in international copyright treaty negotiations and the framing of new legislation being considered in the halls of Congress.[179] The United States delegation to the World Intellectual Property Organization's (WIPO) negotiations to draft a protocol to the Berne Convention regarding copyright in digital works sought to establish provisions that would substantially limit reverse engineering by prohibiting the import, manufacture, or distribution of any device that would circumvent any anticopying technology.[180] The provision ulti-

[178] *See* Karjala, *A Coherent Theory for the Copyright Protection of Computer Software and Recent Judicial Interpretations*, 66 U. CIN. L. REV. 53 (1997).

[179] *See* Samuelson, *The U.S. Digital Agenda at WIPO*, 37 VA. J. INT'L L. 369 (1997).

[180] *See* U.S. DEPARTMENT OF COMMERCE, INFORMATION INFRASTRUCTURE TASK FORCE, INTELLECTUAL PROPERTY AND THE NATIONAL INFORMATION INFRASTRUCTURE: THE REPORT OF THE WORKING GROUP ON INTELLECTUAL PROPERTY RIGHTS (1995) (NII White Paper).

mately adopted was considerably weaker than the U.S. proposal.[181] Nonetheless, the Clinton administration and content interests continue to push for the enactment of implementing legislation that would achieve the same results that they sought, but failed to achieve, in the WIPO treaty itself.[182]

Even if these lobbying efforts to alter copyright law do not succeed, the limitations of traditional copyright law may be of little significance as a result of new opportunities to circumvent copyright law through technological and contractual means.[183] A recent decision of the U.S. Court of Appeals for the Seventh Circuit brushed aside claims of copyright preemption in enforcing a shrink-wrap license that prohibited the copying of uncopyrightable material.[184] This decision, in combination with increased use of highly restrictive licensing agreements, threatens to supplant copyright law with private intellectual property protections that could possibly prohibit reverse engineering of interface specifications and other network features of computer

[181] Article 11: Obligations Concerning Technological Measures

> Contracting Parties shall provide adequate legal protection and effective legal remedies against the circumvention of effective technological measures that are used by authors in connection with the exercise of their rights under this Treaty or the Berne Convention and that restrict acts, in respect of their works, which are not authorized by the authors concerned or permitted by law.

WIPO, Provisional Treaty on Protection of Literary and Artistic Works, art. 11.

[182] *See* H.R. 2281, 105th Cong., 2d Sess. (1995); Letter of intellectual property professors to Honorable Howard Coble, Chairman of the House Subcommittee on Courts and Intellectual Property questioning proposed bill (Sep. 16, 1997) http://www.dfc.org/issues/wipo/profltr.html (visited April 15, 1998); *see generally* Cohen, *Some Reflections on Copyright Management Systems and Laws Designed to Protect Them,* 12 BERKELEY TECH. L.J. 161 (1997).

[183] *See* Samuelson, *Will the Copyright Office Be Obsolete in the Twenty-First Century?,*13 CARDOZO ARTS & ENT. L.J. 55, 60–61 (1994).

[184] *See* ProCD, Inc. v. Zeidenberg, 86 F.3d 1447 (7th Cir. 1996); *see also* Hill v. Gateway 2000, Inc., 105 F.3d 1147 (7th Cir. 1997).

software.[185] In addition, new technological controls on software products—such as new copy protection devices and encryption envelopes—could further limit or regulate the use of software products.[186]

Lax standards, over breadth, and inconsistency in the issuance of patents on software-related inventions pose additional threats to competition and innovation in network markets. Since the late 1980s, the Patent and Trademark Office, with the encouragement of the U.S. Court of Appeals for the Federal Circuit, has relaxed the standards governing the issuance of patents for software-related inventions.[187] Because of the lack of an adequate prior art database, few examiners experienced in the software field, and vague guidelines, the PTO has issued many patents of questionable validity and breadth.[188] In one prominent case, the PTO withdrew, of its own accord, a patent that claimed to cover substantially all multimedia works.[189] Nonetheless, tens of thousands of software patents have been issued and many software engineers, firms in the industry, and scholars have expressed serious concern about how the new regime will play out.[190] The first cases assessing the validity of the wave of software patents recently issued are just reaching the courts now and at least one

[185] *See generally Symposium, Intellectual Property and Contract Law in the Information Age: The Impact of Article 2B of the Uniform Commercial Code on the Future of Transactions in Information and Electronic Commerce*, CAL. L. REV./BERKELEY TECH. L.J. (forthcoming 1998).

[186] *See* Schlachter, *The Intellectual Property Renaissance in Cyberspace: Why Copyright Law Could Be Unimportant on the Internet*, 12 BERKELEY TECH. L.J. 15 (1997).

[187] *See generally* R. MERGES, PATENT LAW AND POLICY 77–119 (1997).

[188] *See* Markoff, *supra* note 52; Galler testimony, *supra* note 52; *League for Programming Freedom*, *supra* note 52; Kahin, *supra* note 52.

[189] *See U.S. Patent Office Throws Out Problematic Compton Patent*, DAILY J., Mar. 28, 1994, at 6.

[190] *See* Galler testimony, *supra* note 52; Kahin, *supra* note 52; Samuelson, *Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer-Related Inventions*, 39 EMORY L.J. 1025 (1990).

early decision was careful to scrutinize the validity of the patents at issue.[191] But given the tens of thousands of software patents that have been issued in recent years, many more tests of the reliability of the patent system remain. Even apart from the problem of bad patents being issued, the exclusivity of patent rights, without any fair use doctrine (apart from the much more limited doctrine of misuse), poses a serious threat to the realization of network externalities.[192]

In view of the economic and social significance of network markets now and in the future, it is essential that the rules governing competition and innovation in these industries be guided by sound policy and not backroom political deals, circumvention of fundamental intellectual property principles through uncritical expansion of contractual freedom, or weak administration of the patent law. The application of traditional copyright doctrines comports with the growing body of economic and legal scholarship relating to network technologies. In order to insure the continued growth of these vital industries, law and public policy should continue to respect these principles.

[191] *See* Quantel, Ltd. v. ADOBE Systems Inc., 1997 U.S. Dist. LEXIS 16908 (D. Del. 1997) (jury verdict rejecting infringement action seeking $138 million in damages); *U.S. Software Maker Fends Off British Patent Claims*, NAT'L L.J., Nov. 3, 1997, at A9.

[192] *See* Menell, *supra* note 13, at 1364–67, 1371.