COPYRIGHT PROTECTION OF COMPUTER DOCUMENTS, REVERSE ENGINEERING, AND PROFESSOR MILLER

Dennis S. Karjala*

I. INTRODUCTION

Initial reservations as to the appropriateness of applying copyright protection to functional works gave way in the 1970s to the view that computer programs, as literary works, particularly in their source-code forms, need protection from piracy. Because copyright law protects against copying, copyrights were a reasonable choice for the protection of programs, especially in view of the deficiencies of patent and trade secret law. Copyright also provided automatic transnational protection of programs under the international conventions. As a result, computer programs are nearly universally the subject of copyright protection.

In the 1990s, however, the debate continues to rage over the proper place of computer programs within the copyright protective scheme. The debate is not over copying of code for resale or for the purpose of creating a usable second program to accomplish the function intended by its author. Copying of this type constitutes copyright infringement, and copyright laws work well as a legal prohibition of that kind of piracy. The debate, rather, is over what other aspects of program technology, beyond literal code, should be protected by the copyright in the program. Academic as well as judicial attention has focused on three problems: the scope of copyright protection in a program; the copyright protection of interfaces, especially user interfaces; and the reverse engineering of programs.

Dramatic judicial developments occurred in each of these three areas in 1992. While the specific subject of this Symposium is reverse engineering, the three problems are intimately intertwined with each other. All three problems raise the fundamental question of the extent to which traditional distinctions between copyright and patent protec-

---

* Professor of Law, Arizona State University, B.S.E. 1961, Princeton University; M.S. 1963, Ph.D 1965, University of Illinois (Urbana); J.D. 1972, University of California (Berkeley).

tion can and should be maintained in the digital age. As a basic policy issue, there is no single "right" or "wrong" logical resolution. Nevertheless, it is an issue that is fundamental to the entire intellectual property protection scheme and should therefore be addressed directly with arguments based on intelligible axioms and logic. Unfortunately, this policy issue is easily hidden in the abstract analysis and terminology of copyright law as applied, often superficially, to specific factual settings.

Professor Arthur R. Miller of Harvard University recently made an important contribution to the ongoing debate with his article reviewing the development of copyright protection for programs in the United States. Professor Miller's article includes a lengthy critique of *Sega Enterprises Ltd. v. Accolade, Inc.*, which held that the reverse analysis of a program, under certain conditions, constitutes a noninfringing fair use. Professor Miller is comfortable with bringing broad aspects of software technology, beyond program code, under the copyright umbrella, but he never explicitly attempts to justify the fundamental policy switch from patent protection to copyright protection. Professor Miller's position represents only one side of the policy debate. Moreover, his arguments, even in traditional copyright terms, are often seriously flawed. Due to Professor Miller's individual prominence and the publication of his article in the preeminent law journal of this country, it is likely that other academics, as well as lawyers, judges, and legislators, will give it serious consideration.

This Article, therefore, attempts to identify and critically analyze the policy lapses and argumentative errors in Professor Miller's article. Rather than do this in isolation, however, this Article first presents a frame of reference by articulating at least one view of the relationship between patent law and copyright law and its implications for the copyright protection of computer programs. This approach addresses the basic policy issue concerning the patent/copyright boundary and offers a reasoned basis for resolving the three major problems mentioned above. Against that framework this Article then undertakes a detailed critique of Professor Miller's article.

II. THE DISTINCTION BETWEEN PATENT AND COPYRIGHT

I suggest that the fundamental difference between patent law and copyright law is simple: Patent law protects creative but *functional* in-

---

3. 977 F.2d 1510 (9th Cir. 1992).
4. This section is based on Dennis S. Karjala, Reverse Engineering of Computer Programs and the Challenge to the Patent/Copyright Boundary, which was presented at the Stockholm
vention; copyright law protects creative but nonfunctional authorship. "Creativity" does not distinguish between the two regimes. Both types of laws protect intellectual creativity, but neither protects, in general, creativity more than the other, nor are the two regimes distinguishable by the type of creativity they protect.

The definition of "functionality" is key to this claim, and the Copyright Act's definition of a "useful article" is the best starting point. A work is "functional" if it performs some utilitarian task other than to inform, entertain, or portray an appearance to human beings. "Usefulness" in the ordinary sense does not make a work functional.

5. Professor Friedman's contribution to this Symposium suggests a different approach to the distinction between patent and copyright subject matter, namely, that the copyright paradigm is applied when copying is easy, easily recognized, and independent invention unlikely, while the patent paradigm is applied when copying is expensive, hard to recognize, and independent invention is likely. David D. Friedman, Standards As Intellectual Property: An Economic Approach, 19 U. DAYTON L. REV. 1109 (1994). This approach essentially treats copyright as a misappropriation law and is supported by the decision to include computer program code under copyright. Moreover, it may be sufficient for providing an economic analysis of the issues raised by standards, which is Professor Friedman's goal. Crucial to that argument, however, is its willingness to consider another "box" for new kinds of intellectual property. If we were free to create new misappropriation statutes at will, we might well decide to protect certain types of functional works under a copyright-like statute, but for a shorter term or with some form of compulsory licensing so that the work in question could be improved by others. Cf. J.H. Reichman, Legal Hybrids Between the Patent and Copyright Paradigms, ___ COLUM. L. REV. ____ (forthcoming 1994). The Semiconductor Chip Protection Act is one case in which this has been accomplished. Each statute would be tailored to the characteristics of the work protected under it and the distinctions between the different subject matters would be multidimensional. For a more detailed analysis of how such tailoring might take place in general, see Rochelle C. Dreyfuss, A Wiseguy's Approach to Information Products: Muscling Copyright and Patent Into a Unitary Theory of Intellectual Property, 1992 Sup. Ct. REV. 195, 221-29.

In fact, however, we rarely write new statutes to deal with new kinds of works. At present we have, in general, only two boxes—patent and copyright. The decision to place a type of work in, for example, the copyright box, brings with it not only protection against literal copying but also prohibitions on compulsory licensing, a very long period of protection, and an uncertain scope of protection against even nonliteral copying, not to mention several hundred years of copyright jurisprudence. For that reason, the courts have not divided patent and copyright subject matter solely according to the scheme suggested by Professor Friedman. The accounting system and implementing forms at issue in Baker v. Selden, 101 U.S. 99 (1879), were at least easy to copy and copying was easily recognized. Nevertheless, copyright protection was denied both for the system and for its "user interface," the implementing forms. The refusal of the Supreme Court to recognize a copyright-like protection for the "photocopying" of boat hulls, Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141 (1989), also shows that there is more to the actual implementation of the patent/copyright distinction by the statutes and the courts than is captured by the features suggested by Professor Friedman. This article concedes below, see infra text accompanying notes 18-21, that basing subject matter distinctions on functionality is not perfect, either. Nevertheless, it is impossible to capture the actual distinction between patent and copyright, as the two statutes are applied today, without regard to functionality.

6. Cf. 17 U.S.C. § 101 (1988). "A 'useful article' is an article having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information."
under this definition. Thus, a map under this definition is nonfunctional, even though it is often quite useful in travelling from A to B. A doll or toy airplane is nonfunctional even though either may be useful in keeping children productively occupied. A picture is nonfunctional even though it may be useful for decorating a house or office. A building, on the other hand, is functional, notwithstanding the fact that part of its design may be to portray an aesthetic appearance to human beings, because it also keeps the rain off our heads while sleeping or working. A computer program in object code form is functional because it causes a computing machine to operate so as to achieve a certain result. This Article claims that the differences between patent and copyright are largely attributable to patent law's primary role as the protector of functional works and copyright law's primary role as the protector of nonfunctional works.

The underlying social policy goals of patent protection and copyright protection are similar. The job of both is to strike a balance between providing an incentive for the creation of works desired by society, recognizing in some fair and just way the efforts of their creators, and insuring a broad public domain that permits later inventors and authors to build on the existing foundation to advance technology and culture for the overall benefit of society. Nevertheless, the copyright and patent statutes effect this social policy balance in very different ways: Patents must be narrowly claimed and protection is narrowly limited to the claim. Patents must be approved by administrative authorities, must involve a "nonobvious" engineering advance, and remain valid for seventeen years. Copyrights, on the other hand, come into existence automatically with no requirement that the rightholder specify precisely which aspects of his work are protected and which are not protected. Moreover, the copyright-protected work need only be the intellectual product of its author, with at most minimal artistic "creativity," and copyright protection continues for roughly seventy-five to one hundred years. In sum, patent protection is narrower than copyright protection, harder to secure, and persists for a shorter period than copyright protection.

The policy basis for these differences is the social desirability, indeed necessity, of allowing later technological creators—creators of functional works—to build on and improve the earlier works of others. Technologically improved products are often substantially similar to the products they improve and therefore the improved product would infringe if the copyright paradigm were applied. Yet, unless the im-

PROTECTION OF COMPUTER DOCUMENTS

1994]  

provement adopts all of the elements of a patent claim, it does not infringe. Current authors of copyright-protected works also build on public domain works and unprotected elements of protected works, but in taking from the latter, the broad scope of copyright protection forces the authors to rely only on the more general or abstract features of the protected works.

Later inventors thus can apply a "ground up" approach to reliance on protected works that precede their own work. Once the inventor is outside all patent claims he is safe. Later authors, however, must adopt a "top down" approach. As their reliance on prior works becomes more detailed they take a serious risk of being held liable for infringement. The functionality/nonfunctionality distinction between patent protection and copyright protection explains these different approaches to infringement. The social utility of allowing subsequent authors to make minor variations on a copyright-protected novel is minimal. For works of fiction, art, and music, variety is the spice of both legal and real life. Society prefers to have one hundred different war novels than one hundred versions of "War and Peace" that differ only in their final chapter. Consequently, the broad scope and long duration of copyright protection for novels fulfill the goal of recognizing the author's creativity without unduly hindering later authors or depriving society of desirable works.8

Technology, however, improves incrementally as later inventors add a bell or a whistle to an earlier invention to make it more desirable or useful to consumers. Many improvements on existing products are rather straightforward, or "obvious" in the sense of patent law, and they are given no intellectual property protection once they are released to the public. Such products often show at least as much intellectual creativity as many copyright-protected works. Their creators, however, have a monopoly only for the period of time that is required for competitors to recognize the value or popularity of the improved product, discover its "secret," and gear up for production and marketing. In the case of technological products, society has drawn the social policy balance at a different point than it has for traditional works of authorship. Society believes that to grant intellectual property rights in ordinary engineering advances would hinder the development of more products than it would encourage. Hence the "nonobviousness" requirement of patent law, as well as its shorter term of protection and its requirement for an explicit statement of the claimed invention, furthers the social

8. Keper-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."
policy goal of promoting both original and improved technological products.

That patent law protects functional works is expressly reflected in the "usefulness" condition that has always been a part of patent law. Copyright law, however, tends to define its subject matter by the types of works protected, such as "literary works," "musical works," and "pictorial, graphic, and sculptural works." If something fits within one of these categories, or is similar to works within these categories, copyright protection is often recognized without any inquiry into whether the work is functional. Nevertheless, the traditional works protected by copyright law are useful only in the sense that they entertain (by presenting an appearance or sound) or inform human beings. Such works can be put to use only through human intervention and interpretation. Thus, the functional/nonfunctional distinction between traditional patent protection and copyright protection has long existed, though it is only implicit in the copyright statute.

United States law has perhaps been one of the clearest in seeking to maintain the distinction between patent and copyright and to limit copyright protection in functional works. First, under the idea/expression distinction, courts narrow the "expression" found in functional or utilitarian works, so works are afforded only "thin" protection. Second, for over one hundred years Baker v. Selden and its progeny have taught that copyright law's reproduction right in a particular work is subordinated to free competition when its assertion would otherwise inhibit competition in the market for non-copyright-protected products. The copyright in a book, for example, does not extend to sys-

9. 17 U.S.C. § 102(a) (1988) (stating that copyright subsists in "original works of authorship" and going on to include enumerated categories within the term "works of authorship").
10. Thus, a recipe for baking a cake might be very "useful" if one wishes to have dessert after dinner, but the recipe itself is not functional because it does not bake the cake. Only the human being reading the recipe does that, and he or she is free to make variations over which the recipe has no control.
12. See, e.g., Continental Casualty Co. v. Beardsley, 253 F.2d 702 (2d Cir.), cert. denied, 358 U.S. 816 (1958). The legal forms involved in this case are functional within this Article's definition, because, in addition to providing information, they establish legal relationships among the parties who execute them.
tems and procedures that are described in the book. Moreover, third parties have always been permitted to copy an unprotected three-dimensional boiler or dress, even though the reverse engineering of any complex product normally involves making intermediate two-dimensional designs based on the product being copied. These cases stress the importance of insuring that copyright law, with its low threshold of eligibility for protection, does not displace the more stringent requirements for protection of functional works under patent law.

To be sure, to the extent that copyright law has protected industrial designs and architectural works, part of the patent/copyright dichotomy based on functionality may have broken down, even before the advent of computer programs. In the Copyright Act of 1976, the United States sought to continue the subordination of copyright protection to function for industrial designs through its separability test for pictorial, graphic, and sculptural works (PGS works). The recent ad-

14. See, e.g., Affiliated Enter., Inc. v. Gruber, 86 F.2d 958 (1st Cir. 1936) (promotional scheme protectible only by patent, regardless of how good it is or how costly it was to develop); Affiliated Enter., Inc. v. Gantz, 86 F.2d 597 (10th Cir. 1936) (similar promotional scheme); Brief English Sys., Inc. v. Owen, 48 F.2d 555 (2d Cir.), cert. denied, 283 U.S. 858 (1931) (only patent is available to protect a new system of shorthand); Arica Inst., Inc. v. Palmme, 761 F. Supp. 1056 (S.D.N.Y. 1991) (method of describing psychological traits not protected); Kepner-Tregoe, Inc. v. Carabio, 203 U.S.P.Q. 124 (E.D. Mich. 1979) (no monopoly on pedagogical technique involved in management training program); see also Pamela Samuelson, Computer Programs, User Interfaces, and Section 102(b) of the Copyright Act of 1976: A Critique of Lotus v. Paperback, 6 High Tech. L.J. 209, 226-27 & n.73 (1991).


16. See Amicus Brief, supra note 11, at 157 (the primary author of this portion of the brief was Professor Reichman). Therefore, when new designs are drawn up based on the three-dimensional product, they are likely to be substantially similar to, and indirectly taken from, the plaintiff's copyright-protected design documents.


19. 17 U.S.C. § 101 (1988) (definition of pictorial, graphic, and sculptural works). This definition states that the design of a useful article is a PGS work only to the extent that it incorporates PGS features that are separately identifiable from and capable of existing independently of the utilitarian aspects of the article. The separability test has caused considerable problems in the
dition of the separate class of "architectural works" to the statutory list of copyright-protected works, however, means that building designs are no longer subject to the separability test. Presumably, the long-standing doctrine of Baker v. Selden will continue to limit or deny copyright protection to functional aspects of buildings, but only time and the courts will tell us. In addition, over the years courts have afforded copyright protection to occasional isolated examples of functional works, in the sense used herein, although these courts rarely evince an understanding that they have actually brought functionality under copyright law.20

The functionality/nonfunctionality distinction between patent law and copyright law is, therefore, not historically perfect. It is surely correct to say, however, that copyright law has generally eschewed protection of function, and judicial and legislative efforts to allow copyright law to control markets for functional products have almost always met with fierce debate.21 Patent law and copyright law have evolved alongside one another over a period of several hundred years, and yet their protection schemes are very different. There must be some reason, other than inertia, that the two have coexisted for so long. Only functionality explains this dichotomy in any fundamental way. To the extent the distinction between the two primary intellectual property regimes based on functionality is not quite descriptively correct, this Article simply offers it as a normative proposition: With its low threshold of eligibility and long protection period, copyright law should not be allowed to trench upon the traditional domain of patent law to protect functional aspects of works absent a clearly articulated social policy basis. In any event, this fundamental policy question should be ex-

courts, but for the purposes of this Article it is enough to note that the effort was made to subordinate copyright to function for this class of works.

20. A clear example is that of standardized test questions that seek to measure intellectual or psychological traits from human responses to the questions. See Educational Testing Serv. v. Katzman, 793 F.2d 533, 539 (3d Cir. 1986); Applied Innovations, Inc. v. Regents of the Univ. of Minn., 876 F.2d 626, 636 (8th Cir. 1989). Even those skeptical of the value of some of these tests will agree that their purpose is to measure real-world phenomena, and therefore they have a utilitarian purpose (measurement) other than to entertain or inform. Therefore, they are functional in the sense used in this Article. Some blank forms also receive protection notwithstanding their functionality as systems for presenting information. Kregos v. Associated Press, 937 F.2d 700, 709 (2d Cir. 1991). See generally Dennis S. Karjala, Copyright and Misappropriation, 17 U. DAYTON L. REV. 885, 922-26 (1992).

21. Professor Reichman describes at length the debates over the inclusion of industrial design within copyright. See generally Reichman, supra note 18. As another example, after the House of Lords decision in British Leyland Motor Corp. v. Armstrong Patents Co., [1986] R.P.C. 279, the British copyright statute was amended to deny copyright infringement in making an article to the design of a design document or model and to include "must-fit" and "must-match" exceptions to the design right in an article. Copyright, Designs and Patents Act of 1988, Arts. 51 & 213(3)(b).
expressly addressed by those seeking broader copyright protection for software: Why should we protect aspects of creative software technology under copyright law while leaving all other creative technological advances to their fate under patent or trade secret law?

In fact, there exists a social policy basis for protecting computer programs, which are functional works, under copyright law. That policy, however, also clearly indicates how copyright protection in programs should be limited to avoid upsetting the careful balance between copyright and patent law that has evolved over time.

III. PROTECTION OF COMPUTER SOFTWARE UNDER COPYRIGHT LAW

Computer software is technology—the means by which computing machines are caused to perform their jobs. As outlined above, patent law is the traditional intellectual property vehicle for protecting technology. Computer programs, communication protocols, hardware-to-software and software-to-software interfaces, as well as many user interfaces, are intrinsically functional in the sense herein defined. They cause computing machines to achieve their results and either serve as or open the doors and windows by which programs are used or interact with each other or with hardware. Thus, computer programs have utilitarian purposes other than to inform or entertain human beings. In fact, except possibly for the purpose of setting “traps” to detect copying, everything in a program and its nonuser interfaces (as well as much of its user interface) is intended by its author to achieve a functional purpose, such as accuracy, speed of processing, optimal use of available resources, facilitating or hindering access, or ease of learning or use. Why did society suddenly turn to copyright law for the protection of computer programs?

Calling programs “literary works” is simply another way of phrasing the question. One could equally, perhaps even more aptly, call programs “methods of machine design,” in that they take a universal machine and transform it into one that achieves a particular result. The real reason a copyright scheme for computer programs has emerged is that many programs are simply the result of technologically straight-

---


23. Because user interfaces may include fanciful, nonfunctional aspects (such as video game characters), it is often important to distinguish them from other interfaces and communication protocols. However, the term “communication protocols, hardware-to-software and software-to-software interfaces” is long and clumsy. Henceforth this Article refers to them as “nonuser interfaces.”
forward application of well-known programming principles to a well-defined problem. Such programs do not meet the requirement of traditional patent law for a nonobvious advance in the art.\textsuperscript{24} Yet, once these programs are distributed in object-code form, they can be copied at low cost in large numbers. Without some form of protection, one should expect computer programs to be under produced. Because the evil to be avoided was slavish copying, because copyright law protects at least against \textit{that}, and because a computer program formally fits the statute's broad definition of a "literary work,"\textsuperscript{26} copyright law became a natural candidate for the protection of programs, notwithstanding their inherent functionality.\textsuperscript{28}

Maintenance of perspective is important, however. The policy basis for bringing functional computer programs under the copyright umbrella is the prevention of slavish copying of code that is ineligible for patent protection. The goal is \textit{not} to protect all creativity found in programs. Other nonpatented works of technology may be freely copied, modified, and improved, no matter how creative the technology is. The creators of such technological developments enjoy only the limited monopoly resulting from the lead time their products have in the market before they can be successfully produced and marketed by competitors. Such copying is permitted, even applauded,\textsuperscript{27} not because society devalues technological creativity, but rather because technology advances incrementally. To forbid such copying would inhibit more creativity than it would engender. Program code arguably requires a different kind of protection, because to allow verbatim electronic copying of programs would reduce the creator's lead time monopoly almost to zero, and that level of protection seems too little.

\textsuperscript{24} At least one commentator has argued that implementing a well-defined process in a programming language is always obvious and therefore nonpatentable. See generally Gary Dukarich, \textit{Patentability of Dedicated Information Processors and Infringement Protection of Inventions that Use Them}, 29 \textit{JURIMETRICS} J. 135 (1989). Moreover, computer programs "as such" are barred from patentability under Article 52 (Appendix I) of the European Patent Convention, although it is not entirely clear just what "computer programs as such" means. J. Betten, \textit{Patent Protection for Software}, in ECIS Symposium entitled "An Emerging World-Wide Consensus on Software Protection," Brussels, Belgium, April 27, 1993.

\textsuperscript{25} "Literary works' are works... expressed in words, numbers, or other verbal or numerical symbols or indicia..." 17 U.S.C. § 101 (1988).

\textsuperscript{26} See generally \textit{New Protectionism}, supra note 7; see also, Dennis S. Karjala, \textit{Copyright Protection of Computer Software in the United States and Japan, Part I}, 13 \textit{EUROPEAN INTELLECTUAL PROPERTY REV.} 195 (1991) [hereinafter Copyright Protection I]; Dennis S. Karjala, \textit{Copyright Protection of Computer Software in the United States and Japan}, 13 \textit{EUROPEAN INTELLECTUAL PROPERTY REV.} 231 (1991) [hereinafter Copyright Protection II]. Another advantage of copyright is the immediate international nature of the protection under the copyright treaties. \textit{Copyright Protection I}, supra, at 196.

The software protection goal, therefore, should be protection against piracy—methods of copying that too greatly upset the traditional balances of legal and nonlegal protection available for works of technology—and not a wholesale revamping of the intellectual property protection scheme for functional works or for technological creativity. This, at least, is the conservative approach—the approach that least disrupts the traditional intellectual property protection balances, especially the delicate balance between copyright law and patent law. If there are grounds for affording even broader protection to program technology under copyright law, they do not arise out of traditional copyright law itself (notwithstanding the formal classification of computer programs as literary works), because traditional copyright law does not protect function. Beyond the vulnerability of code to piracy, no convincing argument has yet been made that distinguishes computer programs from other technological products and leads to a broad scope of copyright protection. Because broad copyright protection for technology is the radical, rather than the conservative deviation from traditional norms, the burden of proving a convincing policy basis for broad copyright protection of functionality should be on those seeking it.

Two further points should be made explicit. First, copyright protection of the set of statements or instructions that constitute computer programs—that is, source and object code—is in itself important and significant protection. Protection of code makes direct copying either for sale or for simultaneous use by others (within, for example, a given business) illegal. Of course, not all instances of program copying can be detected, any more than all copying of other types of copyright-protected works can be detected. No matter what the law says, some people will make copies of useful programs borrowed from their friends without paying. Sales of unlawful copies in any significant amount, however, are readily detectable. Moreover, any employer who distributes illegally made copies of a program among employees in the business runs a serious risk that at some point a disgruntled employee will turn him in. Therefore, protection of program code alone is of great importance. It is simply incorrect to say, as did a committee report for the European Parliament during the debates on the European Software Directive, that if reverse analysis were permitted "legal protection for computer programs would virtually cease to exist."

Second, the Copyright Act makes explicit use of its "useful article" definition to limit the scope of protection in PGS works. The definition of PGS works in section 101 establishes a separability test that the design of a useful article must meet if that design is to qualify for copyright protection. Such design is included within the definition of a PGS work only to the extent it incorporates PGS features that can be identified separately from and are capable of existing independently of the utilitarian aspects of the article. Because computer programs are literary rather than PGS works, the separability test applicable to PGS works does not apply to computer programs.

This limitation of the separability test to PGS works does not affect the functionality limitations on the scope of copyright protection in computer programs. To apply the separability test to computer programs would defeat the whole purpose of including programs within copyright. Program code is wholly functional, so to separate all functional aspects of code from a program would leave nothing to protect. Yet, it is clear that Congress intended to protect program code. That the separability test applicable now to PGS works does not apply to programs says nothing, however, about how the larger functionality doctrine that finds its genesis in Baker v. Selden applies to noncode aspects of programs. This doctrine, which is partially codified in sections 102(b) and 113(b) of the Copyright Act, applies to all classes of functional works. The recent program cases make clear that this general doctrine, abjuring copyright protection of function except for program code, remains intact.

Given that literal code is and should be copyright protected, this Article now turns to a brief consideration of the three problems that are the subject of heated debate: (1) How much of a program beyond literal code, if any, should be protected; (2) Should the protective scheme for the program also cover its interfaces, including its user inter-

terfaces, and if so to what extent; and (3) To what degree, if at all, should reverse analysis be permitted.\textsuperscript{38}

A. The Scope of a Program Copyright

The scope-of-protection problem continues to confound the courts, even after recent judicial efforts to pull back from the expansive and much criticized approach of Whelan, Inc. v. Jaslow Dental Laboratory, Inc.\textsuperscript{35} The courts all know about the "idea/expression" and now the "process/expression"\textsuperscript{34} distinctions, and they are getting more precise concerning the unprotected noncode program elements that must be filtered out in the infringement analysis.\textsuperscript{36} No court, however, has quite come to grips with the fundamental question of why any nonliteral program aspect that somehow survives the filtering procedure can be deemed "expressive." Every aspect of a program, literal and nonliteral, is intended by its author to serve the functional purpose of causing the program to perform in some optimal manner in relation to the constraints set by the environment in which it is to be used (including the nature of its intended or expected users). Many courts and commentators seem to think that the merger doctrine applies in these circumstances. If a wide variety of ways to achieve optimal program performance exists, or if different programmers working independently of one another would likely arrive at somewhat different solutions, any given solution should be deemed "expression."\textsuperscript{38}

The difficulty with the merger approach to idea/expression in functional works is that Baker v. Selden and section 102(b) of the

\textsuperscript{32} For a more extended discussion, see generally New Protectionism, supra note 7; Copyright Protection, supra note 26; Recent Developments, supra note 22; see also Amicus Brief, supra note 11.

\textsuperscript{33} 797 F.2d 1222 (3d Cir. 1986), cert. denied, 479 U.S. 1031 (1987).

\textsuperscript{34} Steven R. Englund, Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs, 88 Mich. L. Rev. 866, 901-03 (1990); see also Gates Rubber Co. v. Bando Chem. Indus., 9 F.3d 823 (10th Cir. 1993).

\textsuperscript{35} Gates Rubber Co., 9 F.3d 823 (ideas, processes, facts, public domain information, merger material, and scenes a faire material, including hardware standards and mechanical specifications, software standards and compatibility requirements, computer manufacturer design standards, target industry practices and demands, and computer industry programming practices); Computer Assoc., 982 F.2d 693 (elements necessarily incidental to program function, including elements dictated by efficiency; elements required by external factors, including elements dictated by mechanical specifications, compatibility requirements, computer manufacturer design standards, demands of the industry served, and widely accepted programming practices; and public domain elements).

\textsuperscript{36} This was the approach of Whelan, which the Tenth Circuit in Gates Rubber approved after cutting back on Whelan's overly narrow conception of "idea." 9 F.3d at 834. See also Englund, supra note 34, at 903 (arguing for protection if the same process can be implemented efficiently by combining and redividing module functions in a number of substantially different ways).
Copyright Act mean much more than the merger doctrine. The systems and processes described in a copyright-protected work are unprotected no matter how many other possible systems or processes may exist to accomplish the same result and regardless of whether they accomplish that result less, equally, or more efficiently. The Court in *Baker* did not inquire into whether other accounting methods existed to accomplish the goals of Selden's system—indeed there are millions—let alone any that were better than the one at issue. The fundamental notion is that functional works not meeting the stringent requirements for a seventeen-year patent must be allowed to develop through incremental change, via the contributions of many creative persons and not just the person who first arrives at a particular stage.

Courts seeking expression in nonliteral program elements should therefore understand that they are fundamentally deviating from copyright tradition. Some people undoubtedly think that such a policy change is desirable, and possibly Congress will someday issue such a directive through legislation. In fact, however, Congress has not yet spoken on the scope-of-protection question for programs. Courts should, therefore, require the proponents of radical change to demonstrate either its clear desirability as a matter of social policy or that there is no other way to interpret the statute and still give meaning to the congressional directive to include programs under copyright law.

Both the definition of a computer program and the apparent concern of the National Commission on New Technological Uses of Copyrighted Works (CONTU) with piracy of literal program elements

---

37. *E.g., Affiliated Enter., Inc. v. Gruber*, 86 F.2d 958 (1st Cir. 1936) (promotional scheme protectible only by patent, regardless of how good it is or how costly it was to develop); *Affiliated Enter., Inc. v. Gantz*, 86 F.2d 597 (10th Cir. 1936) (similar promotional scheme); *Brief English Sys., Inc. v. Owen*, 48 F.2d 555 (2d Cir.), *cert. denied*, 283 U.S. 858 (1931) (only patent is available to protect a new system of shorthand); *Arica Inst., Inc. v. Palmer*, 761 F. Supp. 1056 (S.D.N.Y. 1991) (method of describing psychological traits not protected); *Kepner-Tregoe, Inc. v. Carabio*, 203 U.S.P.Q. 124, 130 (E.D. Mich. 1979) (no monopoly on pedagogical technique involved in management training program); *see Samuelson, Computer Programs, User Interfaces, and Section 102(b) of the Copyright Act of 1976: A Critique of Lotus v. Paperback, 6 High Tech. L.J. 209, 226-27 & n.73 (1991).*

In *Russell v. Northeastern Pub. Co.*, 7 F. Supp. 571 (D. Mass. 1934), the plaintiff had written a book on auction bridge play and charged that defendant had copied a hand therefrom for its newspaper column on bridge (with very minor changes in cards irrelevant to the key point of the instruction). The court found it "clear" that plaintiff had no exclusive right "in the particular distribution of the 52 cards," even though there are billions of such card combinations. The reason, of course, is that the point of instruction is not protected, and the particular hand was a convenient way of making it, although surely not the only one.

38. 17 U.S.C. § 101 (1988) ("computer program" defined as a set of statements or instructions to be used in a computer to bring about a certain result).”

39. The National Commission on New Technological Uses of Copyrighted Works (CONTU) Report accepts the traditional idea/expression dichotomy, but every concrete infringement example offered involves direct and literal copying, such as photocopying of printed source
suggest an interpretation that gives meaning to the congressional directive to protect programs under copyright law. At the same time this minimizes the degree to which traditional copyright law must be bent out of shape to accommodate computer programs. This conservative approach protects program code under copyright law against literal copying and slavish transformations or translations, whether direct or indirect, by modern digital means. This result can be achieved by drawing the "abstractions line" between idea and expression at or near the level of literal code, analogous to the "thin" copyright protection that is afforded to histories, biographies, rule books, legal forms, and scientific works. Adoption of this conservative approach would simply end the debate over whether and how nonliteral program elements can be "expressive."

40. E.g., Harper House, Inc. v. Thomas Nelson, Inc., 889 F.2d 197 (9th Cir. 1989) (utilitarian aspects of "organizers"); Landsberg v. Scrabble Crossword Game Players, Inc., 736 F.2d 485 (9th Cir.), cert. denied, 469 U.S. 1037 (1984) (game strategy); Miller v. Universal Studios, Inc., 650 F.2d 1365 (5th Cir. 1981) (historical report); Hoehling v. Universal City Studios, Inc., 618 F.2d 972, 980 (2d Cir.), cert. denied, 449 U.S. 841 (1980) (historical theory); Rosemont Enter., Inc. v. Random House, Inc., 366 F.2d 303 (2d Cir. 1966), cert. denied, 385 U.S. 1009 (1967) (biography); Continental Casualty Co. v. Beardsley, 253 F.2d 702 (2d Cir.), cert. denied, 358 U.S. 816 (1958) (legal form); Kepner-Tregoe, Inc. v. Carabio, 203 U.S.P.Q. 124 (E.D. Mich. 1979) (outlines for problem solving). The courts thus find all "expression" in these works in the literal and near-literal elements. This policy choice allows the courts to avoid getting bogged down in the otherwise wholly metaphysical question of whether nonliteral structural or theoretical elements that may be needed by later authors attempting to solve a similar problem constitute "expression." It also eliminates the need to address the thorny fact issues that a quality standard (how many ways are there to achieve the same result "equally well") necessarily generates.

41. That is, at least until someone demonstrates a plausible ground for distinguishing program design technology from other technological designs for the purpose of bringing it under copyright instead of patent and trade secret law.
B. Interfaces

Limiting the scope of protection to literal code, including direct or indirect mechanical or electronic translations, also leads to a solution of the second problem—that of the copyright protection of interfaces. Interfaces lie at a level of abstraction well above code. The interfaces are, in fact, essentially part of that "certain result" that the set of instructions accomplishes. They are the doors and windows through which users and other hardware and software make use of the program. The program designer must decide what kind of interfaces are desired and then write code to implement those decisions. Drawing the idea/expression line near literal code means that the program's interfaces would not be covered by the copyright in the program itself. Any copyright protection afforded to an interface would, therefore, have to come from an independent copyright in the interface itself.

As the abstract channels through which hardware and other programs interact with the given program, nonuser interfaces probably do not constitute copyright-protected works at all. It is true that a given nonuser interface can be viewed as a kind of combination of signals, codes, timing events, and the like, and strong protectionists will undoubtedly want to call that a creative "compilation." That approach would stretch copyright subject matter well beyond the breaking point. Every useful article, such as an automobile, can be thought of as a collection of features (radio, electric windows, leather seats, clock and the like), but only the most covetous copyrightist would say in public that any given collection of such features in a particular car would be copyright protected. Nonuser interfaces, moreover, are wholly functional. They do not communicate to human beings so they neither inform nor entertain human beings. Moreover, non-user interfaces are not programs, which constitute the only functional work any of whose functional aspects Congress has expressly brought within copyright law. Any intellectual property protection afforded to nonuser interfaces should come from patent or trade secret law rather than copyright law.

Fanciful aspects of user interfaces, such as some features of screen displays, can be protected as aspects of audiovisual works without doing violence to traditional copyright notions. The real question for user interfaces, therefore, is whether functionality at the user

42. See Dennis S. Karjala, Interfaces, in SOFTWARE INFORMATION CENTER (TOKYO). PROCEEDINGS OF THE SECOND INTERNATIONAL SYMPOSIUM ON LEGAL PROTECTION OF COMPUTER SOFTWARE 629 (1989) [hereinafter Interfaces].

43. There is a meaningful sense, however, in which interfaces constitute program languages. Copyright Protection I, supra note 26, at 199.
level—including user lock-in and standardization—is or should be copyright protected. United States law is well armed to eliminate functional aspects of user interfaces from copyright protection under the tradition of Baker v. Selden, although the two most recent and articulate authorities on the question are in hopeless conflict.

C. Reverse Analysis of Programs

As discussed above, intellectual property law represents a balance between competing policies favoring protection, such as basic fairness to creators of socially desirable works and incentives to their creation, and policies favoring nonprotection, such as the free flow of knowledge and ideas and the social advantages resulting from one creator's building on the work of another. Because this balance must be struck differently for functional works in comparison with nonfunctional works, patent law and copyright law effect their respective protective schemes in very different ways. Copyright law does not give the copyright owner complete control over all possible uses of the protected work, nor does copyright law prohibit use of the protected work in general. In particular, ideas contained in the program are not protected nor, under section 102(b) of the United States Copyright Act, are systems, processes, concepts, principles, or methods of operation.

44. It is useful in this context to consider the ordinary typewriter keyboard. Does the keyboard arrangement simply inform the typist which keys to punch, or does it become, with practice, a part of the instinctive way a human being actually uses the machine? Omitting questions of efficiency, the arrangement of the keys is largely arbitrary. Yet, if a single manufacturer managed somehow to garner the lion's share of the initial market, users who became used to that arrangement would be reluctant to change to a different keyboard. They would be even more reluctant to switch if the first keyboard were efficient for ease and speed of typing, because competitors could offer no inducement to compensate for the difficulties involved in learning a new keyboard. A copyright in the keyboard arrangement—the predigital user interface—would then result in a very long quasi-monopoly in the first popularly used keyboard, in turn resulting in a monopoly on the machines themselves. See New Protectionism, supra note 7, at 45-46. Although some keyboard arrangements were patented, traditional copyright would have denied protection on functionality grounds, and the absence of any reports concerning the assertion of copyright claims in typewriter keyboard arrangements suggests that there was general understanding and acceptance of this result.

45. 101 U.S. 99 (1877); see supra notes 11-17 and accompanying text.


47. Ralph S. Brown, Eligibility for Copyright Protection: A Search for Principled Standards, 70 MINN. L. REV. 579, 588-89 (1985). "The right to control the use of a work, although granted to inventors, has never been part of copyright except as performance may be considered 'use.' Indeed, the absence of a use right helps to justify the relatively casual approach to granting copyright . . . ." Id.
Yet, when a program is available only in magnetically or electronically encoded object code, these theoretically unprotected elements of programs are not readily extractable, because no human being can examine and understand a complex program in this form.\textsuperscript{48} Knowledge of these unprotected elements is often a vital aspect of creating interoperable programs and systems. Moreover, a fundamental part of the copyright balance—especially in view of long term protection—is that ideas and other unprotected elements in publicly available copyright-protected works should be free for all to use and build upon. All other forms of literary works exist in human intelligible form. Anyone is free to take the unprotected elements from a publicly distributed work without the permission of the author for the purpose of creating new works, even competing works. For this reason, reproduction of the work for the purpose of extracting unprotected elements was never an issue for traditional (nonfunctional) literary works.

Computer programs, however, cannot be "read" like books. In order for a human being to understand the ideas and other unprotected elements contained in the object-code form of a computer program, a technological process known as "reverse analysis" is necessary. Part of this process involves making a technical copy or translation of the program through what is called "decompilation" or "disassembly." The question for copyright law is whether such technical copies infringe, even where no use is made of the copies other than to examine them for the purpose of extracting their copyright-unprotected elements.

It is worth noting that, in contrast to traditional works, a program creator has \textit{no} legitimate complaint against reproduction of his source code if he is assured that the source code will not be reinput and used in another computer.\textsuperscript{49} The only value in putting object code into

\textsuperscript{48} Even if the electronic representation is written out in 0's and 1's—which in any event would of course constitute a "copy" of the program—it is extremely difficult if not impossible even for a skilled programmer to make sense of it. See generally Amicus Brief, supra note 11. For an excellent description of the entire reverse engineering process for programs, and the difficulties involved in effecting reverse engineering, see Andrew Johnson-Laird, Technical Demonstration of "Decompilation," 16 COMPUTER L. REP. (Nov. 1992); see also Andrew Johnson-Laird, Reverse Engineering of Software: Separating Legal Mythology from Actual Technology, 5 SOFTWARE L.J. 331 (1992).

\textsuperscript{49} New Protectionism, supra note 7, at 91; Dennis S. Karjala, Protection of Computer Programs under Japanese Copyright Law, 8 EUR. INTELL. PROP. REV. 105, 109 (1986); see also M. Lehmann & Thomas Dreier, The Legal Protection of Computer Programs: Certain Aspects of the Proposal for an (EC) Council Directive, 6 COMPUTER L. & PRAC. 92, 95 (1990). These authors use similar reasoning to conclude, in Germany prior to the EC /Software Directive and in the absence of a general fair use provision, that courts should treat interim copies made in the course of reverse analysis as mere technical copies and not as infringing reproductions. They argue that the copyright goal is to assure the author adequate participation in the economic exploitation of her work. Because decompilation for the purpose of extracting unprotected elements is not the form of exploitation envisioned by the prohibition against copying, a technical copy made in the
human-readable form is to permit study of the program either for making
other programs or for making more effective use of the program in
question. Computer programs are the technology for using computers
and limitations on source code distribution can seriously impede the
flow of technological knowledge concerning computer program develop-
ment and use. Program creators naturally want to use copyright law to
prevent study of their source code, because copyright protection effec-
tively maintains their monopoly not only over the copyright-protected
elements of their programs but also over unprotected ideas and meth-
ods. The desire of program creators for more protection than copyright
law has ever given to any other form of technology, however, is no rea-
son to provide such protection.50

The basic argument in favor of permitting reverse analysis of pro-
grams for the purpose of extracting and using unprotected elements has
been made in several forums51 and has now been strongly confirmed by
both Atari Games Corp. v. Nintendo of America, Inc.52 and Sega En-
terprises Ltd. v. Accolade, Inc.53 It is therefore unnecessary to go into
the argument in depth. A few words, however, on factor two of section
107's fair use factors, are appropriate. It is primarily this factor that
distinguishes the fair use analysis relating to programs in object-code
form from that for other types of copyright-protected works.

First, computer programs are inherently functional works that di-
rectly cause machine processes to be performed.54 Because computer
programs constitute the means by which useful processes are performed
in a computer, there is a risk that overprotecting a program could give
the copyright owner a patent-like monopoly on the program's function-
ality—on what the program does—without the showing required to ob-
tain a patent. This Article has argued that computer programs have
properly been brought under the copyright umbrella, notwithstanding
their inherent functionality, because their special vulnerability to
piracy in the form of electronic copying distinguishes them from other
works of technology. If the scope of copyright protection is properly

---

50. Dennis S. Karjala, The First Case on Protection of Operating Systems and Reverse
Engineering of Programs in Japan, 10 EUR. INTELL. PROP. REV. 172, 176 (1988) (hereinafter
Reverse Engineering).

51. See generally Amicus Brief, supra note 11; LaST Frontier Conference Report on Copy-

52. 975 F.2d 832 (Fed. Cir. 1992). The author has served as a consultant to counsel and
potential expert witness for Atari Games in this litigation.

53. 977 F.2d 1510 (9th Cir. 1992).

54. This should be contrasted with maps or recipe books, which merely convey useful infor-
mation to humans.
limited to insure that technological ideas remain free for other programmers to use, the law should not allow indirect copyright protection of such ideas by blindly applying copyright concepts and language appropriate to traditional (nonfunctional) works.

Second, the object-code form in which computer programs are publicly distributed is not amenable to study, research, or even comprehension except through the making of a copy intelligible to human beings. If decompiling publicly distributed but human-incomprehensible object code is considered infringement, the public never receives its fair part of the copyright bargain: a seventy-five-year limited monopoly in exchange for the opportunity to read and be inspired by the work and a free license to copy all of the ideas and processes embodied in the program.

All other copyright-protected works carry their ideas and other unprotected elements on their face. Anyone is free to use these unprotected elements without the permission of the author, even in works that directly compete. No fair use question arises because it is not infringement simply to extract unprotected elements. Computer programs in object-code form are alone among all types of publicly distributed copyright-protected works in being unreadable by human beings. Unless the program can be put into an intelligible form, it cannot be studied as a whole. Without studying the program as a whole, its unprotected ideas and processes cannot be discerned, nor can unprotected elements embedded in programs, such as interface information, be detected as such. One of the principal justifications for the very long period of copyright protection, the absence of general compulsory licensing provisions, and the automatic subsistence of protection without any independent review of novelty, is that the elements needed by later authors to advance cultural development by building on the past are precisely those that are unprotected by the copyright. If access to unprotected elements in publicly distributed works can be technologically denied and enforced through copyright law, the fundamental copyright equation is thrown out of balance.

Having set out a particular policy-based framework for the copyright protection of computer programs, largely based on recognizing

55. See supra notes 33-41 and accompanying text.
57. Of course, legal access to the work is usually necessary in order to extract ideas or other unprotected elements without violating some other legal rule, but legal access is always available when a work has been publicly distributed.
and maintaining the traditional boundary between patent law and copyright law, this Article now turns to an analysis of the approach to software protection taken by Professor Miller.

IV. Critique of Professor Miller's Approach to Reverse Analysis

Analysis of Professor Miller's approach to the copyright protection of computer programs must be effected on at least two levels. In many ways he is most vulnerable precisely at the technical level of argument he purports to adopt, namely: the application of traditional copyright law to what he regards as no more than yet another technological development in a long line that historically have been successfully brought under the copyright umbrella. At this level the errors and omissions in the argument are relatively concrete and, therefore, more easily grasped and evaluated.

Criticism at that level alone, however, does not reach the fundamental area of disagreement between this Article's approach and that of Professor Miller. In order to fully evaluate Professor Miller's position, one must consider the policy question posed at the outset: To what extent should functionality in computer programs, beyond literal code, be protected under copyright instead of (or in addition to) patent and trade secret, the traditional modes of protection for technological creativity? This Article has argued that the conservative approach relies on the long history of mutually coexisting, but separate, intellectual property regimes to handle new technologies. In other words, there should be a presumption in favor of the traditional division of labor until concrete problems of either over- or under-protection appear that the traditional regimes seem incapable of adequately accommodating. Program code, which is wholly functional, is such an example. In bringing program code under copyright protection, albeit for good reasons, one should be cautious in using the copyright mode for other aspects of program technology.

Professor Miller, on the other hand, is not troubled by the protection of functionality, as defined herein, under copyright law. He does not even seem to regard it as a serious issue, as he never addresses the question of functionality in terms of the differences between patent law and copyright law and why or whether those differences should be maintained. Nevertheless, implicit views on the copyright protection of

58. Miller, supra note 2.
59. Miller, supra note 2, at 982-83.
60. See supra text following note 1.
61. See supra text accompanying notes 22-31.
62. See supra text accompanying note 6.
function permeate his analysis and, because they involve fundamental value choices, they cannot be said to be wrong. Law and legal argument are rarely perfectly logical. If Professor Miller's value choices are in fact representative of the community, what appears to be erroneous or incomplete in his technical argument may simply represent healthy growth in the law's adjustment to the information age.

The basic policy issue, then, is one over which reasonable people will differ. Still, in evaluating or accepting more technical arguments concerning the application of copyright law to software, it is important to understand what underlying policy is being implemented. Those inclined to accept Professor Miller's views at the policy level are likely to be less troubled by deficiencies at the more technical level of argument. Those who feel, on the other hand, that the stringent requirements for patent protection were created for a reason having continuing force will resist statutory interpretations resulting in automatic protection for technological creativity that has not been shown to constitute more than an ordinary engineering advance. This is especially so when protection endures for the long period provided by copyright, with no compulsory licensing and a substantial similarity test for infringement that prevents incremental improvements by others. For such people, error or incompleteness in the technical argument constitute a good ground for its rejection. It is therefore important to isolate the value choices implicit in the technical arguments.

Professor Miller's failure directly to address the policy choices inherent in his approach makes it difficult to challenge those choices or even to identify them outside the context of his technical argument. For this reason, this Article attempts to identify the technical flaws in his approach and to consider them sequentially along with their underlying, if implicit, policy choices. This section begins with aspects of Professor Miller's argument that have implications beyond, but nonetheless relevant to, the reverse analysis topic that is the subject of the Symposium. It then considers in some detail his approach to the reverse engineering problem.

A. **Assimilation to Other Copyright-Protected Works**

Professor Miller compares and finds similarity in the ways computer programs and traditional literary works are created: Both require "imagination, originality, and creativity," involve individuality, and are often produced under constraints.63 While this is true, it fails to distinguish between works that are traditional patent subject matter and those that have been copyright-protected. Many technological works,

---

63. Miller, supra note 2, at 983-84, 1010.
from airplanes to electronic circuitry to drugs, involve much imaginative intellectual creativity. It is true that no two programmers would independently design a program with the same structural details, just as no two novelists would independently compose the same detailed plot.\textsuperscript{64} It is equally true, however, that no two watchmakers would independently choose all the same parts or put them together in the same way. If programmers are constrained by the need for communicative precision and musicians by the limited ranges of instruments and voice,\textsuperscript{65} so too are all other technological products that are unprotected by copyright law. Airplane wing designers, for example, are constrained by the qualities and cost of the materials available to them and the intended use and range of the airplane. Technological solutions to problems that successfully optimize a product under constraints can often be described as "elegant" or "aesthetically pleasing." That the solutions call for such description, however, has never been a ground for taking them out of patent protection and into copyright law.\textsuperscript{66}

Both patent law and copyright law protect intellectual creativity. Works that have no function other than to inform, entertain, or portray an appearance to human beings are the traditional stuff of copyright. Works that do have a function other than to inform, entertain, or portray an appearance to human beings are traditional patent subject matter. The works causing the greatest traditional challenge to this division of labor are industrial designs and architectural works, which function both to portray an appearance to humans and to perform a task other than to inform or entertain. Comparisons based on creative content of works or the way in which works are produced, without reference to functionality, can never explain why some creativity falls under patent

\textsuperscript{64} Miller, supra note 2, at 984.

\textsuperscript{65} Miller, supra note 2, at 984, 1010.

\textsuperscript{66} The very title of the recently aired three-part series on Public Television on developments in materials science—"The Stuff of Dreams"—illuminates this point. Here, materials researchers are compared to artists in the way they work, creating new materials that are "stronger, harder, lighter, and smarter" than any we have seen before, in some cases "literally atom by atom." In Part 2, a researcher interested in better boatmaking states that the current approach is not to adapt existing materials but to decide what is needed and then to fabricate the material for the purpose. How is this intellectual process different from that of a programmer who decides (or is told) what is needed and designs a program for the purpose? Should we on this ground and without further policy analysis protect these creative new materials under copyright? In Part 3, scientists are compared to painters and actors in their striving to create. Another scientist states, "To be creative you have to ask yourself . . . the right question . . . and it's true for an artist who's doing a painting, it's true also for the sculptor who's trying to show something in three dimensions, to show feelings, and it's true also for the scientist." The series closes with these words about materials scientists: "Without their dreams there is no progress, no true science. We wait and watch as they stretch their imaginations and artistry, like Michelangelo with his marble, Matisse with his cutouts, or Aeschylus with his quill." Imaginative or artistic creativity simply does not distinguish between patent and copyright subject matter.
protection, some under copyright protection, and some under neither (like the theory of relativity). Computer programs (as opposed to some user interfaces), creative though they may be, have only a utilitarian function other than to inform, entertain, or portray an appearance. Explanation for their inclusion under copyright protection must be sought elsewhere than in the creativity involved in producing them.

B. Intent of Congress and of CONTU

Professor Miller makes numerous references to the intent of Congress and of CONTU, the Commission appointed by Congress to investigate the problem of software protection. His article is a kind of crescendo of increasingly strong assertions about the congressional intent relating to computer programs. He states, for example, that an underlying premise of both Congress and CONTU was that programs are entitled to copyright protection under the same principles that govern other literary works. A few pages later he refers to the desire of Congress “to accord computer programs full copyright protection” and to Congress’s “intended incentives and rewards.” His next claim is that CONTU and Congress rejected arguments for exceptions to copyright

67. Professor Miller does compare computer programs to architectural works with his claim that both combine aesthetic and utilitarian elements. Miller, supra note 2, at 988. With respect to user interfaces, this may be a valid observation if human users judge the aesthetics of a screen display in ways similar to their aesthetic judgments of buildings or industrial designs. The comparison, however, is misleading for program code or structure as well as nonuser interfaces. While programmers might refer to a particular program structure or block of code as “elegant” or “aesthetically satisfying,” terms of this type are widely used in all (non-copyright-protected) design technologies to express admiration for a solution to a problem that optimizes use of the available resources in a new and valuable way. This is not the kind of aesthetics that distinguishes between patent and copyright.

Moreover, even with respect to user interfaces, the comparison to architectural works may be disingenuous. Professor Miller treats interfaces simply as nonliteral aspects of the program, notwithstanding that they are a long way up the scale of abstraction toward that “certain result” effected by the “set of statements or instructions” that the Copyright Act defines to be a computer program. This has a technical advantage for those seeking broad copyright protection for user interfaces, because otherwise they might arguably be subject to the separability test for pictorial, graphic, and sculptural works (“PGS works”). That may also explain why Professor Miller makes the comparison only to architectural works, which Congress took out of the PGS work definition (so the separability test no longer applies to them) and not to industrial designs, which remain subject to the separability test. None of this game-playing with the statutory definitions, however, should take our focus away from the basic issue of how much noncode functionality in programs, program structure, or interfaces should sensibly be included under copyright. If copyright protection of function does more harm than good, Baker v. Selden remains solid authority for denying copyright protection for functional aspects of all of them.

68. This authors’s explanation is the vulnerability of program code to nonproductive, inexpensive, and almost immediate exact copying. See supra notes 23-28 and accompanying text.

69. See supra note 39.

70. Miller, supra note 2, at 1008.

71. Miller, supra note 2, at 1010 n.156.
protection for decompilation and interface copying based on concerns for standardization or access to program ideas. Next he asserts that a copyright monopoly in Sega's popular game entertainment system was part of the incentive Congress intended to provide program authors.

In fact, Congress has never spoken on the scope-of-protection question for computer programs. All we know from either Congress or CONTU is that the idea/expression distinction should be applied to limit the scope of program protection. In fact, CONTU stated its objectives for copyright protection of programs rather clearly:

To provide reasonable protection for proprietors without unduly burdening users of programs and the general public, the following statements concerning program copyright ought to be true:

1. Copyright should proscribe the unauthorized copying of these works.
2. Copyright should in no way inhibit the rightful use of these works.
3. Copyright should not block the development and dissemination of these works.
4. Copyright should not grant anyone more economic power than is necessary to achieve the incentive to create.

By emphasizing the need for case-by-case analysis and failing to provide examples of infringing activity not involving direct copying, CONTU evidently expected courts to consider these four policy goals as technology and judicial knowledge concerning computer programs developed. Professor Miller himself applauds the case-by-case decision-making as long as it is going in the direction of ever broader protection. Only with cases like Computer Associates and Sega does he begin to fear that the "intent of Congress" to provide broad incentives may be thwarted. By protecting program code, however, the courts are fulfilling objective one in CONTU's list. If the courts later interpret the scope of protection or fair use to avoid affording unnecessary eco-

72. Miller, supra note 2, at 1013.
73. Miller, supra note 2, at 1020.
74. For example, the House Report on the 1976 Copyright Act states: "Section 102(b) is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law." H.R. REP. No. 1476, 94th Cong., 2nd Sess. 57 (1976) (hereinafter House Report). CONTU also referred generally to the idea/expression distinction, but came to no conclusions concerning where the line between the two should be drawn. CONTU Report, supra note 39, at 20, 25.
75. CONTU Report, supra note 39, at 12.
76. CONTU Report, supra note 39, at 22-23; see also supra note 39.
77. E.g., Miller, supra note 2, at 1009.
78. 982 F.2d 693 (2d Cir. 1992).
79. 977 F.2d 1510 (9th Cir. 1992).
80. Recall that the definition of a computer program as a set of statements or instructions came from CONTU. CONTU Report, supra note 39, at 12.
nomie power to a program copyright owner or to promote the use, development, or dissemination of programs, they are attempting to put flesh on the analytical framework suggested by CONTU. There is certainly nothing in the CONTU Report, much less any congressional history, to support the kind of broad copyright protection for program structure or interfaces that Professor Miller believes is appropriate.

C. Functionality

Professor Miller sidesteps the difficult aspects of dealing with program functionality by equating the term “useful” with “functional” and pointing out that many traditionally copyright-protected works are “useful”: “[T]oday’s copyright law protects a wide variety of disparate ‘writings,’ including fact compilations, dictionaries, code books, encyclopedias, advertising, and ‘how to’ instruction manuals, that, like many computer programs, have a primarily utilitarian rather than aesthetic, entertainment, or educational purpose.”81 Later even Shakespeare becomes “useful” by providing a sense of satisfaction to readers through his sonnets.82 In fact, none of the listed types of works (or the works of Shakespeare) have an intrinsic utilitarian function other than to portray an appearance or convey thoughts or information.83 Therefore, none of the works are functional in the sense used herein to distinguish patent and copyright subject matter, or in the Copyright Act’s definition of a useful article.84 Professor Miller then errs in the other direction by asserting that computer programs are not useful articles.85

81. Miller, supra note 2, at 986 (footnote omitted).
82. Miller, supra note 2, at 1030 n.244.
83. Fact compilations, dictionaries, encyclopedias, advertising, and instruction manuals manifestly are not “useful articles” under the Copyright Act’s definition. Professor Miller’s reference to code books is presumably to Reiss v. National Quotation Bureau, Inc., 276 F. 717 (S.D.N.Y. 1921), in which a book of meaningless code words was held to be copyright protected. The “useful” article issue is whether such a list of words has an intrinsic utilitarian function other than to portray themselves or to convey information. Although such words do not in themselves convey information, their apparent function is simply to portray themselves when used to make a secret code. The Reiss case does not hold that any specific code—meanings and/or syntax that might be added to these code words by the user—would be protected. Such a code might well be denied copyright protection as a language, that is, a system for conveying information. Had the code book been newly created hieroglyphic-type characters, few would doubt its copyright protectibility, or deny its status as a useful article under the Act’s definition, because the only function of the characters would be to portray themselves (whether or not the user added meanings or syntax). The code words in Reiss portray themselves as letter groups rather than pictures, but they have no other utilitarian function, intrinsic or otherwise. Consequently, they would not be “functional” in the sense that, this Article has argued, distinguishes patent and copyright subject matter. See supra note 6 and accompanying text.
84. See supra note 6 and accompanying text.
85. Miller, supra note 2, at 987 n.41. Professor Miller’s cited authority for this erroneous statement is Lotus Dev. Corp. v. Paperback Software Int’l, 740 F. Supp. 37, 71 (D. Mass. 1990). Judge Keeton makes many errors similar to those of Professor Miller in the Lotus series of opin-
On the basis of his broad definition of functionality, Professor Miller concludes that "functionality poses no per se bar to copyrightability." Clearly he is correct that copyright law has always protected fact compilations and instruction manuals, although of course the scope of copyright protection in such works is "thin." Moreover, Professor Miller is free to adopt his own definition of "functionality." However, by choosing such a broad definition, broader than and different from that found in the Copyright Act itself, he obliterates the crucial difference between computer programs and other kinds of "useful" literary works for the purpose of analyzing the scope of copyright protection that should be afforded to programs. In equating the functionality of a computer program to the "functionality" of a dictionary, and then noting that dictionaries have always been copyright-protected, he implies that both copyrightability as well as the scope of protection can be decided without reference to the fundamental policy issue: How much of program functionality, as defined in a way that distinguishes between patent and copyright subject matter, should be brought under the copyright umbrella and why? Nevertheless, he never says how one distinguishes between what he calls "utility" and what he calls "expression," and questions mount as one reads on further in his paper. He writes that in applying the Computer Associates filters, for example, "a court must employ considerable caution in excluding efficient or speedy..."
program expression lest it undermine the effective protection of computer programs. Dissection of a program rather than comparing competing works in their entirety "runs the risk of reducing it to a collection of unprotectable elements and may ignore the synergism inherent in the combination of its constituent elements." In applying the merger and scenes a faire doctrines,

a court should not declare a computer program element uncopyrightable simply because it can be conclusorily classified as 'efficient' or 'externally determined.' Rather, the court must be careful, in order to protect a program's expressive elements, to define precisely what types of efficiency and what external demands on a program should remain free of copyright.

What does Professor Miller mean by "efficient or speedy expression" in a program? Does he mean that code or structural elements will be protected notwithstanding that they do a job well or fast, as long as there are numerous other equally good or speedy ways of doing it? If so, that presumably means he would deny protection to a major innovative jump in programming technique for efficiency or speed—a jump, in other words, that other programmers could not match without duplicating the essence of the innovative advance. This, however, would put him at odds with Judge Keeton in the Paperback decision. Judge Keeton said that the more innovative the advance, the greater the need for copyright protection. In fact, Professor Miller himself earlier quotes from Paperback in support of his argument that "uncritical application of Altai's language would penalize the most effective (and in some senses the most artistic) programmers." This sounds as if he is equating "effectiveness" with "art," but then what does his separation of "utility" from "expression" mean?

On the other hand, if Professor Miller does believe that major technological advances in programming are copyright protected, then what is the program "utility" that goes unprotected under his analysis? Similarly, when he speaks of "synergism" inherent in a combination of

90. Miller, supra note 2, at 1004 (emphasis added).
91. Miller, supra note 2, at 1006 (emphasis added).
92. Miller, supra note 2, at 1009.
93. Lotus Dev. Corp. v. Paperback Software Int'l, 740 F. Supp. 37, 79 (D. Mass. 1990). "Copyright protection would be perverse if it only protected mundane increments while leaving unprotected as part of the public domain those advancements that are more strikingly innovative." Id. Professor Miller quotes this language with apparent approval later in his article. Miller, supra note 2, at 1020 n.199. Professor Miller also appears to approve another portion of the Paperback opinion rejecting the denial of copyright protection for functional elements for fear that it would inhibit software manufacturers from offering their best products: "Copyright is not a synonym for imperfection." 740 F. Supp. at 58; see also Miller, supra note 2, at 1005 n.132.
94. Miller, supra note 2, at 1005 (emphasis added).
elements, he presumably means that it is something to be protected.\textsuperscript{95} But what is “synergism” if not some better operation of elements so combined that the whole is greater (or better) than the sum of its parts? And why is not that improvement considered “utility?”

There appear to be only two possibilities: Either Professor Miller wishes to protect unique and highly innovative jumps in software technology or he will protect only those lesser advances for which a wide variety of equally good approaches are available. In either case, his approach protects functionality under copyright law. The first possibility does so directly and dramatically, which may be why he does not overtly admit support for it. The second possibility, however, protects functionality as well, although it is more easily camouflaged by reference to the merger doctrine: If there are many ways of doing something equally well, what is the problem with protecting any one of them? That is probably why Professor Miller often hints that he means the latter, without ever rejecting copyright protection for uniquely optimal innovative advances.\textsuperscript{96}

In fact, however, \textit{Baker v. Selden} stands for much more than merger.\textsuperscript{97} The Court did not ask how good or efficient Selden’s accounting system was in refusing to extend the book’s copyright to the system. The system was not copyright-protected because intellectual innovations of that type should be protected, if at all, only after a showing that they meet the stringent requirements of patent law. Technological innovations in any other field are subject to the same requirements, and ordinary advances may be freely copied regardless of how many other equally efficient ways there are of accomplishing the same result.\textsuperscript{98}

There are good reasons for the \textit{Baker} rule of not inquiring into quality to determine copyright protectability. Courts have the advantage of hindsight several years after a technological advance. This makes it easier to see that other avenues were available that did not occur to the original engineer on the job. Moreover, Judge Keeton is correct when he says that protection of the poor or fair, but not the good or best, is perverse.\textsuperscript{99} It places plaintiffs in the awkward position

\textsuperscript{95} He says that dissection of programs runs the risk of missing the synergism, but missing it would not be important if synergism were in any event unprotected.

\textsuperscript{96} For example, he cites with apparent approval cases that distinguish protectible from unprotectible elements by looking at the practical number of ways that existed to accomplish the goal. Miller, \textit{supra} note 2, at 1007 & n.141. He also refers approvingly to \textit{Whelan’s} and Paperback’s analysis that distinguishes between idea and expression depending on whether there are only a few or many ways of accomplishing something. Miller, \textit{supra} note 2, at 996-99.

\textsuperscript{97} \textit{See supra} note 37 and accompanying text.

\textsuperscript{98} Recently, the Supreme Court strongly reaffirmed this point. \textit{See} Bonito Boats, Inc. v. Thunder Craft Boats, Inc., 489 U.S. 141 (1989).

\textsuperscript{99} \textit{See supra} note 93.
of trying to prove that the program they are fighting so hard to defend really is not very good. Moreover, common sense cautions against establishing a system that encourages software producers to hold back their best technological versions of a program on the advice of their lawyers, fearing that the best version is "too good" to qualify for copyright protection. Nevertheless, the solution is not necessarily to put all program technology under copyright law. To the contrary, the solution may be to look at what has worked well for over one hundred years. Traditional copyright law does not distinguish the breadth of its protection according to the intrinsic quality (as opposed to the generic type) of the work or element in question.\(^{100}\) The plot of a potboiler receives the same protection as that of *Huckleberry Finn*. The Theory of Relativity, however, is as wholly unprotected by copyright law as a modest theoretical advance in multibody dynamics. Accounting and shorthand systems enjoy no copyright protection regardless of their quality relative to other such systems. The implication of copyright tradition for computer programs is clear: Protect no noncode elements of programs under copyright law and allow the truly innovative advances in software technology the benefits of patent protection.

Now that machine design is done with words rather than pictures ("literary" as opposed to "graphic" works), Professor Miller would apparently take innovations in those designs outside of the patent paradigm. He almost surely means to protect ordinary innovations in the creation of nonliteral elements of programs, and he probably means extraordinary innovations as well. The latter have always been, and remain today, protectible under patent law.\(^{101}\) The former have traditionally been unprotected, except through trade secret law, not because they are uncreative or undervalued but because, in the long run, society does better by allowing a wide variety of second comers to build on the existing unpatented technological base and make further improvements. In the main, these improvements will also be unprotected from adoption by others. Professor Miller implicitly seeks to change this age-old scheme by protecting not only program code, but also noncode elements of programs, notwithstanding their functionality (or "efficiency,"

\(^{100}\) Bleistein v. Donaldson Lithographing Co., 188 U.S. 239 (1903).

\(^{101}\) To the extent that current interpretations of patent law do not protect "nonobvious" advances in the art of making computer programs, I, at least, disagree with the interpretation but suggests that the solution would be to fix the patent law (or adopt a misappropriation statute, see Reichman, *supra* note 5) rather than provide the protection through copyright. Copyright lasts too long to protect technological advances and its substantial similarity test for infringement prevents both incremental improvement as well as "inventing around" the protected advance. In any event, it appears that patents are readily available today for nonliteral program aspects that meet the patent standards. See, e.g., John P. Sumner & Steven W. Lundberg, *The Versatility of Software Patent Protection: From Subroutines to Look and Feel*, 3 COMPUTER L., June 1986, at 1.
“effectiveness,” or “synergism”). Others may agree with this policy change, but all supporting it should give reasons that directly deal with the policy issue and not pretend that this is simply a straightforward application of traditional copyright law.

D. Programs as Literary Works

Professor Miller makes much of CONTU’s103 supposed “underlying premise . . . that computer programs are entitled to copyright protection under the same principles that govern other literary works.”104 Although he admits, as he must,104 that the scope of protection under traditional copyright law varies according to the type of literary work, he asserts that Sega105 and Atari Games106 contradict this premise by according “weaker” protection to computer programs than to other works.107 Professor Miller also asserts that no “standardization exemption” exists for the category of literary works, concluding therefore that Congress ruled out weighing the public interest in standardization in determining whether decompilation for certain purposes is a fair use.108

Put aside that CONTU refers to programs as literary works only for the purpose of concluding that they are copyright-protected ("copyrightable")109 and that in CONTU’s discussion of the scope of protection there is not a single reference to programs as literary works.110 Put aside as well that in analyzing the scope of protection, CONTU cites only two cases, both holding that the scope of protection in works having functional elements is “thin.”111 Even if CONTU had expressly adopted the premise attributed to it by Professor Miller, the very principle that allows variation in the scope of protection according to the type of work means that certain types of literary works receive “weaker” protection than others. Some courts have explicitly recognized that such “thin” copyrights can be so weak as to be almost non-

102. Professor Miller evidently assumes that this is also Congress’s underlying premise.
103. Miller, supra note 2, at 1008.
104. See supra note 40 and accompanying text.
105. 977 F.2d 1510 (9th Cir. 1992).
107. Miller, supra note 2, at 1022.
108. Miller, supra note 2, at 1021.
110. Needless to say, there is no reference to them as literary works in CONTU’s reverse engineering discussion, because CONTU did not address that problem at all.
existent.\textsuperscript{112} Professor Miller would presumably like to see computer programs treated under copyright law as if they were novels or plays, which receive a broader scope of protection than histories, biographies, rule books, fact works, legal forms, and scientific works.\textsuperscript{118} Simply stating, however, that literary works should all be treated under the same copyright principles does not achieve this goal.

One need not go to policy considerations concerning the copyright protection of functionality to challenge the attempted treatment of programs under copyright law as if they were novels or plays. Even as a matter of technical copyright argument, many would find computer programs more analogous to rule books, recipes, legal forms, or scientific works than to novels and plays. Courts also, of course, distinguish between types of literary works for purposes of fair use, giving greater protection to fictional and artistic works than to works of fact.\textsuperscript{116} That the nature of computer programs as a class of copyright-protected work renders them in certain circumstances (such as reverse analysis)\textsuperscript{118} susceptible to legitimate copying under the fair use privilege does not suggest treatment of computer programs under different copyright principles. Computer programs are treated under the same section 107 factors and other fair use principles applicable to \textit{all} copyright-protected works.

Moreover, the notion that Congress has or has not created "standardization exemptions" for entire classes of works (but not, evidently, literary works) not only contradicts the legislative history of the 1976 Act\textsuperscript{116} but also Professor Miller's own regard for "a fair use doctrine [that] is telephoto in character and highly dependent on the factual circumstances of each case."\textsuperscript{117} In fact, courts have often been concerned with standardization problems in connection with literary works.

\textsuperscript{112} See, e.g., Continental Casualty Co., 253 F.2d at 705 ("such a view [allowing similarity of language necessary to the functional use intended as a legal form] leaves little, if any, protection to the copyright owner; in fact, it comes near to invalidating the copyright") (quoting Crum v. Pacific Mut. Life Ins. Co., 140 F.2d 182, 184-85 (7th Cir. 1944)).


\textsuperscript{115} See supra notes 50-55 and accompanying text.

\textsuperscript{116} See House Report, supra note 74, at 65 (fair use is an "equitable rule of reason" and the criteria for balancing the equities are "in no case definitive or determinative").

\textsuperscript{117} Miller, supra note 2, at 1017.
Usually, however, the elements to be standardized are simply denied copyright protection so the problem is treated as one of scope-of-protection rather than fair use. Paper sizes, binding techniques, and library classification schemes all lie outside the copyright umbrella. When a court denies protection to "terms of art" in a legal form, the court gives effect to standardization concerns and limits the scope of the copyright protection in furtherance thereof. Once a court has concluded that a particular program element lies outside the scope of protection, whether for standardization reasons or otherwise, Professor Miller's denial of a "standardization exemption" in the fair use analysis for literary works, even if it were otherwise correct, becomes irrelevant. The real issue remains whether it is fair use to make intermediate copies or translations of protected programs solely for the purpose of extracting unprotected elements (whether or not related to standardization) and to use those elements in otherwise noninfringing new programs.

E. Copyright and Access

Professor Miller challenges two policy arguments in support of the notion that copying of computer programs should be permitted in order to gain access to the program's ideas: (1) without access to ideas the constitutional purpose of promoting progress in science and the useful arts will be undermined; and (2) copyright protection should not be used to deny access to ideas because copyright is not supposed to protect ideas. Professor Miller's response to the first argument is that other copyright-protected works exist in a format that is unintelligible to many users and is often comprehensible only to an expert. A computer program, he asserts, is entitled to as much protection as a musical score or a literary work written in a foreign language, even though each of these latter works is decipherable by only a small percentage of the public. As to the second argument, he notes that this country's copyright law now protects unpublished works, which means that access to some works may be denied for the entirety of the period they are protected (and, he might have added, even beyond).

Professor Miller's response to the first argument does not address the access problem for programs at all. Although many consumers will prefer to have music performed rather than buy the sheet music, or to read an English translation of a foreign-language book rather than


119. Professor Miller's substantive arguments concerning the role standardization concerns should play in the fair use analysis are considered below. See infra notes 159-64 and accompanying text.

120. Miller, supra note 2, at 988-89.
learn the language, everyone has access to published musical scores and literary works in foreign languages. Musicians who are capable of making use of the ideas in musical scores can take advantage of that access to study the score and learn from it. No copying is needed for them to extract the unprotected ideas for use in creating new and desirable works that the listening public can enjoy. Similarly, other writers who are proficient in the foreign language can buy and read the foreign-language book and, without copying it, extract its ideas for use in creating new works for the benefit of all. Computer programs in electronic form are fundamentally different, because even though they may be widely distributed, not even the experts can study and use the ideas they contain without reverse analysis, and that invariably involves some degree of copying or translation.

Professor Miller's attempt to refute the claim that access to ideas is part of the copyright bargain thus reduces to his second response: Access to unpublished works is often unavailable, and yet copyright protection is extended to such works. In this regard, those who accept the access claim must admit that this response has force. No one, to this author's knowledge, has argued that the general public has an inherent right of access to the ideas in unpublished works simply on the ground that the works are copyright protected. It is clear, therefore, that the access argument for permitting reverse analysis of programs must be limited. The question is whether Congress, by extending copyright protection to unpublished works, intended to remove access to ideas altogether as a part of the copyright analysis of the scope of protection and fair use or intended something less drastic.\(^1\)

Under the 1909 Copyright Act, publication was the dividing line between federal copyright and the so-called state "common law copyright." After labelling the dual system "anachronistic, uncertain, impractical, and highly complicated," the House Report lists four reasons for moving to a uniform federal system: (1) To promote uniformity and avoid the practical difficulties involved in determining and enforcing rights in a variety of forums; (2) To eliminate the increasing artificiality of "publication" as a dividing line after the 20th century communications revolution; (3) To place a time limit on the exclusive rights in unpublished works; and (4) To improve international dealings in copyrighted material.\(^2\)

1. Professor Rice's contribution to this Symposium points out that the issue is not whether access is a condition of copyright protection but rather whether copyright may be used affirmatively to prevent access to otherwise copyright-unprotected elements. David A. Rice, Sega and Beyond: A Beacon for Fair Use Analysis . . . as Far as It Goes, 19 U. DAYTON L. REV. 1131 (1994).

These reasons do not indicate an overarching intent to reformulate the fundamental copyright balances. An author who was willing to forgo the economic returns that normally come only through distribution of her work had the right and power, by not publishing, to exclude or limit access to her ideas prior to the 1976 Copyright Act. The Act’s drafters sought to effect no basic change in that position. There is no indication, however, that the drafters intended to allow authors whose works were widely distributed to preclude subsequent authors from using the ideas in their works by publishing the works in a consumer-useful but human incomprehensible form.

Indeed, if access to ideas, systems, processes, methods of operation, or discoveries, for the purpose of use by later authors in creating new works, were deemed by Congress to no longer be a factor in interpreting the statute, there would be no need for section 102(b) of the Copyright Act. That is, if consumer use of the ideas contained in a work was the sole consideration, that purpose would be satisfied by public distribution of the work. Professor Miller’s position incorrectly assumes that the only public purpose served by access is consumer use of the work in question. In fact, communication to the public and consumer use have nothing whatever to do with the idea/expression distinction or the goals of section 102(b). Access to both idea and expression is provided by public distribution of traditional works, and that access both supplies and justifies the author’s economic reward. The idea/expression distinction and section 102(b) are aimed at defining what later authors can freely use in creating new works for the benefit of the public. Computer programs are unique among all classes of copyright-protected works in that they are usually publicly distributed in a form that cannot be read or understood even by experts until, through the process of reverse analysis, someone puts the programs in a human-comprehensible form. Therefore, the access argument retains its force notwithstanding the decision to include unpublished works within the ambit of federal copyright.

123. William F. Patry, The Fair Use Privilege in Copyright Law 401 (1985). Mr. Patry argues that the public’s need for access to programs is fully satisfied by the marketing of the original, so that section 102(b) plays no further role. Professor Miller quotes this language of Mr. Patry with approval and makes the same error in asserting that the objective of copyright is merely to make creative works available to society, whether or not literal expressions or underlying ideas of those works are directly available to the public. Miller, supra note 2, at 1029 & n.239.

124. Amicus Brief, supra note 11, at 156 n.29.
F. Reverse Analysis

Professor Miller makes an extended argument against the holdings of Sega and Atari Games. A general theme that runs through this argument is that allowing decompilation, apparently for any purpose but certainly for the purpose of reverse engineering to create a functionally equivalent work, results in “diminished copyright status” for computer programs. This section addresses this general theme first and then proceeds to Professor Miller’s more particular concerns.

1. Decompilation and Diminished Copyright Status

Professor Miller is worried that allowing decompilation results in “diminished copyright status,” subordinates the interests of software creators in favor of those who copy for commercial ends, accords “a lower level of protection to the program’s copyrightable elements” in order to facilitate access to unprotected elements, and “impairs the owner's traditional rights under American law.” These general complaints about lower levels of protection assume a kind of absolutist character of copyright’s reproduction right that flies in the face of the fair use notion. They also assume a harm to the program copyright owner simply from the act of copying or translating the program into human-readable form.

In fact, as discussed above, unless a decompiled computer program is recompiled for use in a computer, or unless protected elements from the decompiled program are used in creating a new program, the owner has suffered no copyright-cognizable harm by the translation. Computer programs are made and purchased for use, not for reading. Copying or translating into human-readable form, in itself, does not create a single usable copy that can cut into the copyright owner’s sales. If the use of unprotected elements to create a competing program results in loss of sales, that is simply the result of applying the general principles of copyright law to computer programs in the same way it is applied to all other works. No textbook author has a copyright-cognizable claim for profits lost to a competing work on the ground that

125. Miller, supra note 2, at 1013-32.
126. Miller, supra note 2, at 1013.
127. Miller, supra note 2, at 1020.
128. Miller, supra note 2, at 1021.
129. Miller, supra note 2, at 1032.
130. See supra notes 49-50 and accompanying text.
131. Applying copyright principles uniformly is precisely the sort of thing Professor Miller says, in general, ought to be done.
the second author has “stolen” ideas. The owner of a program copyright has no such claim, either.132

A similar automatic-harm-from-copying complaint seems to be at the heart of Professor Miller’s handling of the third fair use factor—the amount and substantiality of the portion used. He finds copying the entire work in the reverse analysis process “most clearly inconsistent” with fair use and feels that the Sega court unduly minimized such copying by focusing on the much smaller amount of copying in the end product. This, he says, contradicts the court’s earlier conclusion that intermediate copying was infringement and encourages decompilers to go on hunting expeditions through copyright-protected programs.133

In fact, the Sega court made no contradiction. Fair use analysis became necessary only because of the court’s conclusion that intermediate copying was prima facie infringement. Surely Professor Miller would not suggest that infringement analysis and fair use analysis must be identical.134 The whole point of fair use is to permit the courts to avoid rigid application of the statute’s exclusive rights, thus avoiding “stifling the very creativity which that law is designed to foster.”135

132. Professor Miller also worries that protection of trade secrets embedded in computer programs might be impaired if programs can be freely decompiled. Miller, supra note 2, at 1026. He may be correct, but the issue is the extent to which this is a legitimate copyright concern, at least if the embedded trade secrets are not themselves protected by copyright. Owners maintain trade secret protection by keeping them secret, not by releasing them in products from which they can be discerned through reverse engineering. Reverse engineering has always been a legitimate means of discovering technological trade secrets. The only difference between program technology and any other technology is that we have decided to protect programs with copyright. In this author’s view, the conservative approach to the copyright protection of software treats programs as an exception brought into copyright by the failure of the other regimes adequately to protect program code. This approach leaves other technological aspects of programs to be treated like other technologies, unless someone comes up with a principled policy reason for deviating. Professor Miller evidently believes in stronger protection for software technology—using copyright to protect trade secrets, for example—but he should supply reasons for this view that go beyond assertion of abstract copyright concepts.

133. Miller, supra note 2, at 1018.

134. Later, however, Professor Miller says, “In Sega, the fair use exception was allowed to consume the rule against intermediate copying. Because intermediate copying infringes even if the end product does not, it is not apparent how intermediate copying was excused by a contention that the noninfringing nature of the end product made the commercial exploitation only derivative and ‘of minimal significance.’” Miller, supra note 2, at 1019 n.193. Here Professor Miller appears to be saying that because intermediate copying is infringement, making intermediate copies can never be fair use. Again the difference between programs in object-code form and ordinary books resolves his quandary. No one need make an intermediate copy of a book to extract and use its unprotected elements. One must do so with computer programs, however, and it is a fair use—notwithstanding that copying is prima facie infringement—because the end product is desirable and because the copyright owner suffers no copyright-cognizable harm from the practice.

Moreover, the feared "hunting expedition" does no more harm to a program owner's legitimate interests than decompilation in the search of specific information. As long as nothing copyright-protected is taken from a publicly distributed program for reuse in a computer, the copyright owner loses no sales and suffers no other copyright-cognizable harm.

2. Decompilation and the Disguise of Infringement

Professor Miller is concerned not only that decompilation permits reproduction of the entire program by the decompiler—which, as discussed in the previous subsection, in itself does not harm the copyright owner—but also that the decompiler might, after decompilation, "electronically massage the copy until every trace of that illicit reproduction is obscured."\textsuperscript{136} This places, he asserts, "an unprecedented burden on the copyright owner to discover and prove infringement."\textsuperscript{137}

This author fully agrees with Professor Miller that this type of electronic massaging after decompilation should be considered copyright infringement. The disagreement is simply over whether proscribing decompilation would do anything to ameliorate the problem. If all traces of the original have vanished and direct evidence is unavailable, infringement will indeed be difficult to detect. This difficulty in discovering the provenance of the end product is ever present, however, regardless of what the law formally prohibits.

Moreover, if a competitor will falsely deny having engaged in "electronic massage" after decompilation, he is likely to deny having decompiled as well, assuming that decompilation is illegal. In fact, if decompilation is legal for the purpose of extracting and using unprotected elements, many suspected cases of infringement might simply involve an admission of the decompilation activity, as occurred in both \textit{Sega}\textsuperscript{138} and \textit{Atari Games}.\textsuperscript{139} This admission would focus the dispute on the nature of the elements taken and would give the copyright owner some leads concerning how the suspect program was created. The discovery process might then turn up real evidence of "electronic massage," if it occurred, as opposed to independent creation based solely on unprotected elements.

3. Decompilation and Derivative Work Rights

Professor Miller makes another argument that should have been dispelled long ago but somehow perseveres: Permitting decompilation

\begin{enumerate}
\item Miller, supra note 2, at 1026.
\item Miller, supra note 2, at 1027.
\item 977 F.2d 1510 (9th Cir. 1992).
\item 975 F.2d 832 (Fed. Cir. 1992).
\end{enumerate}
somehow denies copyright owners their right to authorize and prepare derivative works. Although Professor Miller is not explicit, he presumably relies on the statutory definition of "derivative work" as a work "based upon one or more preexisting works." The flaw in this claim is clear: although one work may be "based on" a second work in an ordinary language sense, that does not mean it is "based upon" the second work in the copyright sense. Under copyright law a work cannot be a derivative work unless it is based on expressive features of a preexisting work. Otherwise, a work solely "based on" the ideas contained in an existing protected work would necessarily constitute an infringing derivative work, even though it expresses those ideas in otherwise noninfringing language. With the new derivative work definition, Congress evidently intended simply to define the types of works eligible for copyright protection as derivative works under section 103. Congress most certainly did not intend to expand the scope of copyright protection to ideas so as to render section 102(b) a nullity.

4. Decompilation and Adverse Economic Effect on Competitive Production Costs

Professor Miller argues that permitting decompilation allows the creation of a market substitute after others have incurred the risk and expense of development, which he describes as "an especially inappropriate result given the extraordinary discrepancy between the cost of creating the software and the cost of duplicating it." He provides no evidence of his own to indicate an extraordinary discrepancy between the cost of creating the original software and creating functionally identical software that uses only unprotected elements of the original software. Rather, he relies on other authors, all lawyers, who assert

141. See generally Amicus Brief, supra note 11.
142. See 1 Melville B. Nimmer & David Nimmer, Nimmer on Copyright § 3.01 (1991). "If that which is borrowed consists merely of ideas and not of the expression of ideas, then although the work may have in part been derived from prior works, it is not a derivative work." Id.; see also Litchfield v. Spielberg, 736 F.2d 1352, 1357 (9th Cir. 1984), cert. denied, 470 U.S. 1052 (1985); United States v. Taxe, 540 F.2d 961, 965 n.2 (9th Cir. 1976).
143. Some commentators and even a few courts have overlooked this logical absurdity to which a broad interpretation of the "based upon" language in the derivative work definition leads. See, e.g., Duncan M. Davidson, Common Law, Uncommon Software, 47 U. Pitt. L. Rev. 1037, 1095 (1986); John M. Conley & Robert M. Bryan, A Unifying Theory for the Litigation of Computer Software Copyright Cases, 6 Computer L.J. 55 (1985); SAS Institute, Inc. v. S & H Computer Systems, Inc., 605 F. Supp. 816, 828-30 (M.D. Tenn. 1985) (dictum, as the court also held that the activity both violated a license agreement and that the final work itself infringed).
144. See Litchfield, 736 F.2d at 1357 ("[plaintiff's] arguments that section 106(2) radically altered the protection afforded by the law of copyright are frivolous").
145. Miller, supra note 2, at 1026.
that reverse engineering software is much easier and cheaper than reverse engineering other industrial products.\textsuperscript{146}

Analysis of the adverse economic effect of decompilation involves an important factual determination: How much cheaper and easier is it to create a competing program through decompilation followed by use of only unprotected elements than to create a functionally equivalent program wholly independently? One hopes that this Symposium will shed some light on that question. Even starting from unprotected information obtained through decompilation, the competitor must still independently code, debug, test, and market his product.\textsuperscript{147} One colleague at this Symposium, an experienced program developer, has written that reverse engineering does not reduce the cost of marketing competing software—that reverse engineering is in fact difficult and more time-consuming than designing a functional equivalent from scratch.\textsuperscript{148}

Two further points relating to Professor Miller's concerns must be noted. First, the authors on whom he relies to show the ease of reverse engineering, and therefore the adverse economic effect on program copyright owners, do not in fact address the issue of decompilation followed by independent programming using only unprotected elements. Rather, they simply assume that the imitator will try to disguise his copying activity through what Professor Miller has termed "electronic massage."\textsuperscript{149} As discussed above,\textsuperscript{150} formally proscribing decompilation will not assist copyright owners in detecting or proving infringement in these cases. Moreover, when electronic massaging can be detected and proved, it amounts to infringement independent of the decompilation activity. Second, Professor Miller's entire argument on the adverse economic effects of permitting decompilation is based on the assumption that the decompiler wishes to produce a functional equivalent of the target program. This argument is completely inapplicable to decompilation for the purpose of learning information necessary for compatibility. In fact, in both \textit{Sega}\textsuperscript{151} and \textit{Atari Games},\textsuperscript{152} and probably in the majority of decompilation cases, the goal of the reverse analysis activity was to obtain information necessary for inter-operability. Armed with the information necessary to allow their games to run on Sega or Nintendo hardware, the defendants in both cases incurred all the ex-


\textsuperscript{147} See \textit{New Protectionism}, supra note 7, at 78 n.163.


\textsuperscript{149} Lake, supra note 149, at 5.

\textsuperscript{150} See supra paragraph following note 137.

\textsuperscript{151} 977 F.2d 1510 (9th Cir. 1992).

\textsuperscript{152} 975 F.2d 832 (Fed. Cir. 1992).
penses associated with independent program (video game) development. Decompilation provided no development cost advantage over that of the games of the plaintiffs with which their games were competing.153

5. Decompilation and Other Market Impacts

In analyzing the fourth fair use factor—market impact of the use—the Sega court emphasized that prohibiting decompilation for the purpose of learning necessary interface information would allow Sega to monopolize the market for games that would run on the Sega game consoles. This monopoly would run counter to the statutory purpose of promoting creative expression. Professor Miller challenges this reasoning as an argument for standardization that lacks rational or empirical foundation.154 Instead of defining the relevant market as that for Sega game cartridges, he argues, the court could have looked to the market for game consoles and required the defendant, Accolade, to design and sell its own home entertainment system. He supposes that the Sega court chose the market for Sega cartridges instead of Sega consoles as the relevant one because of Sega’s success:

The parallel to a standardization argument is apparent: popular systems are to lose their copyright protection merely because their popularity designates them “standards,” a proposition that would not be taken seriously if the copyrighted works were Steinbeck’s Grapes of Wrath, Hemingway’s The Sun Also Rises, or Miller’s Death of a Salesman.155

In fact, the court’s choice of the relevant market for purposes of copyright law interpretation was perfectly sensible. Copyright gives a monopoly in the copyright-protected work, not in the market for hardware products on which the work is used.156 Professor Miller’s reference to

153. A discussion of interoperability may be the appropriate spot to correct yet another error made by Professor Miller. He correctly quotes Article 6 the European Community Software Directive as permitting decompilation to obtain information “necessary to achieve the interoperability of an independently created computer program with other programs . . . .” Council Directive 91/250/EEC of 14 May 1991 on the Legal Protection of Computer Programs, 1991 O.J.(L122) 42, 45. Professor Miller claims that the Directive does not allow decompilation in order to produce competitive products. Miller, supra note 2, at 1031 n.248. This is simply wrong. The Directive’s language directly covers, for example, decompilation of operating software for the purpose of obtaining interoperability information to create a competing operating system (the “independently created” program) compatible with third-party application software (the “other programs”). Moreover, the legislative history, both before and after formal adoption of the Directive, makes it clear that the Article 6 exemption applies to directly competing programs. Thomas C. Vinje, Interoperable Product Development under the EC Software Directive, 8 COMPUTER L & PRAC. 190, 193-94 (1992).
155. Miller, supra note 2, at 1020.
156. See supra notes 11-17 and accompanying text.
Steinbeck, Hemingway, and Miller again treats all literary works as if they are afforded the same scope of copyright protection. In fact, even under traditional copyright law courts have always distinguished among different classes of works.\textsuperscript{187} Novels and plays are indeed broadly protected, but copyright protection for histories, biographies, scientific and technical works, rule books, and legal forms is “thin,” that is, they are protected essentially only against verbatim copying. If traditional literary work copyright law is to be applied to programs, notwithstanding their sui generis character as objects of copyright protection, the question is whether programs are more analogous to novels and plays or to the latter set of more narrowly protected literary works. Moreover, Professor Miller’s reference to “standards” in the context of novels and plays would be more aptly analogous to video game cartridges if one were looking to the methods of binding or printing such books or to page size, none of which, is copyright protected.

Professor Miller’s argument also incorporates the notion that the Sega “system” somehow loses copyright protection due to its popularity, in disregard of section 102(b) of the Copyright Act which explicitly denies copyright protection to “systems.” More fundamentally, the argument disregards the more than one hundred year-old doctrine, beginning with \textit{Baker v. Selden}, that prevents copyright from indirectly protecting through its reproduction right functional elements of works that are more properly subject to the more stringent requirements of patent law.\textsuperscript{188}

6. Decompilation and Standardization

Professor Miller disputes standardization or inter-operability as a justification for decompilation. He argues, first, that even if there are social benefits from standardization, those benefits should not necessarily be achieved at the expense of copyright owners. He feels that comparable arguments could be made to justify, for example, free rights to translate literary works into other languages to make them more accessible to non-English-speaking persons or to reproduce books for use in schools. Congress has considered these and similar arguments and granted limited exemptions. Professor Miller believes that the courts should not tamper further.\textsuperscript{189} He next argues that standardization is not necessarily beneficial, citing the “QWERTY” typewriter keyboard standard that has been universally adopted notwithstanding its inferi-

\textsuperscript{157} See supra notes 40, 113 and accompanying text.
\textsuperscript{158} See supra notes 11-17 and accompanying text.
\textsuperscript{159} Miller, supra note 2, at 1029-30.
ority to other arrangements.\textsuperscript{160} Finally, he asserts that, where beneficial, appropriate standardization will occur through operation of normal market forces.\textsuperscript{161}

The first argument is the same one considered above: It assumes harm to the copyright owner simply from the fact of decompilation.\textsuperscript{162} When a book is reproduced for use in schools, the author is deprived of the economic return from sales of his work that copyright law is designed to guarantee. Mere decompilation of a computer program, however, does not deprive the copyright owner of any sales because the decompiled form cannot be used in a computer.

Professor Miller's second argument based on the QWERTY keyboard arrangement would seem to be one favorable to fair use. Given that an inferior product has become a de facto standard due to user resistance to learning new techniques, does Professor Miller believe that protecting the QWERTY arrangement, and thereby bestowing a benefit on the creator of an inferior product, would be a good idea? The argument must be that, if QWERTY had been a protected arrangement, other typewriter manufacturers would have been forced to use something different. Some of these different arrangements might have been better than QWERTY and, with time, such arrangements might have taken over the market. If that is his argument, skepticism is justified by the fact that better typewriter keyboards were in fact developed but nobody succeeded in marketing them widely. In any event, we have empirical questions concerning the benefits of standardization in particular circumstances, and the question is how the unavailability of clear empirical data should affect the copyright analysis. If copyright is used the way Professor Miller would evidently prefer, standardization would be impossible except through licensing. That this could result in a monopoly or near monopoly appears evident by the dominance of Nintendo and Sega in the video game hardware arena. On the other hand, fair use can permit what now appears to be healthy competition in the video game market yet remain subject to reconsideration if actual harm can be shown from decompilation or if the same level of competition becomes technologically possible without decompilation.\textsuperscript{163}

\textsuperscript{160} Miller, supra note 2, at 1030.

\textsuperscript{161} Miller, supra note 2, at 1031-32.

\textsuperscript{162} See supra notes 131-33 and accompanying text.

\textsuperscript{163} Compare Williams & Wilkins Co. v. United States, 487 F.2d 1345 (Ct. Cl. 1973), aff'd. 420 U.S. 376 (1975) (photocopying of articles from research journals in national library is a fair use, partly on the basis of the high transaction costs in negotiating permission) with American Geophysical Union v. Texaco, Inc., 802 F. Supp. 1 (S.D.N.Y. 1992) (photocopying by researchers at private profit-seeking company is not a fair use, partly on the ground that the Copyright Clearance Center now reduces the transaction costs to manageable proportions).
Traditionally, United States courts have been careful not to allow copyright's reproduction right to interfere with competition in markets for copyright-unprotected products.\footnote{See supra notes 11-17 and accompanying text.} Decisions like \textit{Sega} only continue this tradition. The same policy basis that denies direct copyright protection to functional compatibility information—that copyright is a bad tool for regulating markets for functional products—supports denial of indirect copyright protection of that same information. We could, of course, allow copyright to regulate such markets much more broadly, but such a policy choice requires policy-based justification.

Finally, the argument that standardization occurs to some extent through voluntary arrangements hardly has relevance to a finely tuned, case-specific fair use analysis. Both the \textit{Sega} and \textit{Atari Games} courts were aware that Sega and Nintendo had offered licenses under licensor-specified conditions. Those conditions allowed the hardware manufacturers to control both the types of games that would be available for their hardware and, at least to some extent, the price. They attempted to maintain this position through technological devices that in any other area of unpatented technology would be freely subject to reverse engineering. The crucial element of the devices—the functional "key" to the lock—is not copyright protected. The fair use analysis in both cases is whether "rigid application" of copyright's reproduction right should be used to limit competition to the conditions of the Sega and Nintendo licenses. That standards in other circumstances might arise from voluntary arrangements does not enter into this analysis.

V. Conclusion

Computer programs are unique as a class of copyright-protected work. Their inherent functionality distinguishes such programs not only from musical and graphic works but also from other works falling within the "literary work" definition of the Copyright Act. Program code has been brought within copyright protection for good reason. Going beyond code to protect functional, so-called "nonliteral" aspects, of computer programs requires policy justification for what would be yet a further deviation from the traditional division of labor between the patent and copyright systems.

Professor Miller evidently believes that much software technology beyond program code can be safely brought under the copyright umbrella. Nearly all of his arguments, however, purport to be straightforward application of traditional copyright law. This article has demonstrated that his arguments, although at times superficially appealing, do not make the case even as a matter of technical copyright interpre-
tation. Nor has he even attempted, much less provided, a policy justifica-
tion for ignoring functionality—the traditional boundary between
copyright and patent—in the analysis. In short, the Sega court got it
right.