

2-1-1998

COPYRIGHT LAW: Copyright Protection of Computer Program Structure

Dennis S. Karjala

Follow this and additional works at: <https://brooklynworks.brooklaw.edu/blr>

Recommended Citation

Dennis S. Karjala, *COPYRIGHT LAW: Copyright Protection of Computer Program Structure*, 64 Brook. L. Rev. 519 (1998).
Available at: <https://brooklynworks.brooklaw.edu/blr/vol64/iss2/3>

This Article is brought to you for free and open access by the Law Journals at BrooklynWorks. It has been accepted for inclusion in Brooklyn Law Review by an authorized editor of BrooklynWorks.

COPYRIGHT PROTECTION OF COMPUTER PROGRAM STRUCTURE*

Dennis S. Karjala[†]

INTRODUCTION

It has long been settled that copyright plays an important role in the protection of computer software. Beyond the protection of program code,¹ however, the scope of copyright protection in a computer program remains a matter of heated debate among commentators and a matter of extreme confusion in the courts. This judicial confusion is nicely exemplified by the Second Circuit's 1997 decision in *Softel, Inc. v. Dragon Medical & Scientific Communications, Inc.*² The Second Circuit, in *Computer Associates International, Inc. v. Altai*,³ had earlier established what has become the most widely accepted test for separating protected from unprotected elements in computer programs. The *Softel* panel,

* ©1998 Dennis S. Karjala. All Rights Reserved.

[†] Professor of Law, Arizona State University. I am indebted to Professor Leo Raskind for his many helpful comments on an earlier draft of this article.

¹ Code is the specific listing of instructions or commands that the computer is to execute. Most actual program writing is accomplished in "source code," a high-level language whose commands do not have a one-to-one correspondence with the commands actually executable by a given machine. Source code is, however, much easier for human beings to work with than the actual set of machine language instructions. In order to bring the program into useable (executable) form, the source code must be translated into machine language ("object code") that can be represented to human beings as the famous binary 0s and 1s. In executable form, however, code must exist in the computer's memory as wholly unreadable physical signals (such as a higher voltage for a "1" and a lower voltage for a "0") in order for the machine to execute the program. Dennis S. Karjala, *Copyright, Computer Software, and the New Protectionism*, 28 JURIMETRICS J. 33, 36-38 (1987) (hereinafter *New Protectionism*). See also the discussion in *Whelan Associates, Inc. v. Jaslow Dental Lab., Inc.* 797 F.2d 1222, 1230-31 (3d Cir. 1986). It is well established that both object code and source code are protected by the copyright in the computer program. *Computer Assocs. Int'l, Inc. v. Altai*, 982 F.2d 693, 702 (2d Cir. 1992). Source and object code are often referred to as "literal code" and exact subsets thereof as "literal elements" of the computer program. Other elements of the program, such as structure sequence and organization ("SSO"), are sometimes labelled the "non-literal elements" of the program.

² 118 F.3d 955 (2d Cir. 1997), cert. denied, 118 S. Ct. 1300 (1998).

³ 982 F.2d 693 (2d Cir. 1992).

however, understood neither the correct technical application of the *Computer Associates* test nor its implicit underlying policy basis. As a result, the case resorted to metaphysics to determine what is protected "expression" in a computer program and what is not. Indeed, much of the language of the *Softel* opinion harkens back to the approach of the Third Circuit in *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*,⁴ which was expressly rejected in *Computer Associates*.⁵ Thus, *Softel* is not a step forward but may, in fact, represent a retrogression along the bumpy road to a coherent system of intellectual property protection for computer software.

In this Comment I analyze *Softel* in some detail and explain why the decision is in conflict with the policy underlying the *Computer Associates* approach. I also explain how the *Softel* court substituted metaphysics for analysis by looking to case authorities involving traditional works, such as novels and factual compilations, and by ignoring the technological nature of computer software. Recognizing that software is, in fact, technology—the technology for using computers—is crucial for placing the analysis of copyright scope on firm ground. The technological nature of computer programs distinguishes them from nearly all types of traditional copyright-protected works. To determine the scope of protection by analogy to novels or compilations is inapt and leads to results that are undesirable as a matter of social policy. Yet to treat them as unique types of copyright subject matter—which they are—requires a policy-based standard for separating protected "expression" from unprotected "idea." I have written many articles in recent years explaining the appropriate standard and how it is derived from a coherent interpretation of intellectual property law as a whole, especially the relationship between copyright and patent.⁶ The *Softel* decision provides an opportunity to reiterate some of that analysis in an attempt to illustrate how the *Softel* panel could have better handled the problem before it.

⁴ 797 F.2d 1222 (3d Cir. 1986).

⁵ 982 F.2d at 705-06.

⁶ See, e.g., Dennis S. Karjala, *A Coherent Theory for the Copyright Protection of Computer Software and Recent Judicial Interpretations*, 66 U. CIN. L. REV. 53 (1997) (hereinafter *A Coherent Theory*); Dennis S. Karjala, *Copyright Protection of Computer Programs, Reverse Engineering, and Professor Miller*, 19 U. DAYTON L. REV. 975 (1994) (hereinafter *Reverse Engineering and Professor Miller*); *New Protectionism*, *supra* note 1; Dennis S. Karjala, *The Relative Roles of Patent and Copyright in the Protection of Computer Programs*, ___ JOHN MARSHALL J. COMP. & INFO. L. ___ (forthcoming 1998) (hereinafter *Relative Roles of Patent and Copyright*).

The doctrinal copyright question at issue in cases like *Softel* is what constitutes protected "expression" in a computer program and what constitutes unprotected "idea."⁷ The closest the *Softel* opinion gets to a standard for distinguishing the two is the notion of merger or, more precisely, the absence of merger.⁸ That is to say, the court reasoned that if a variety of possibilities exists for accomplishing something through programming technology, any one of those specific possibilities is protected expression.⁹ Merger, however, is inadequate as a standard for separating idea from expression in nonliteral elements of computer programs because it fails to take into account the policies underlying the exclusions from copyright protection contained in section 102(b) of the Copyright Act.¹⁰ With functional works like computer programs, we must analyze such policies carefully, following the principles of *Baker v. Selden*,¹¹ to

⁷ The term "idea" here is used as a shorthand for all original but copyright-unprotected elements contained in a work. Section 102(b) of the Copyright Act, 17 U.S.C. § 102(b) (1994), states, "[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work."

⁸ Merger is a copyright doctrine that denies protection to what would otherwise be protected expression when the idea or concept being expressed is amenable to a single, or only a few, meaningful modes of statement. See *Morrissey v. Procter & Gamble Co.*, 379 F.2d 675 (1st Cir. 1967). As such, it exists as a *limitation* on the protection of expression that otherwise would be copyright protectable. It is not a doctrine that is designed to specify in the first instance what elements of a work constitute its expression and what elements constitute its idea. As Richard Stern has pointed out, to use absence of merger—the existence of a number of ways of accomplishing an object—as a basis for inferring expression fallaciously inverts the logic of *Baker v. Selden*, 101 U.S. 99 (1879), and *Morrissey*, 379 F.2d at 675. Richard H. Stern, *Copyright in Computer Programming Languages*, 17 RUTGERS COMPUTER & TECH. L.J. 321, 368-69 (1991); see also *infra* notes 63-66 and accompanying text.

⁹ See 118 F.3d 955, 965-66 (2d Cir. 1997), *cert. denied*, 118 S. Ct. 1300 (1998). Here the court concluded that the alleged use of the same fifteen "functionally identical" commands in Dragon's programs and the use of menus and external files to cause the commands to cue modules of code created a colorable claim that there was expression in *Softel*'s design. See *id.* at 966. The court goes on to say that "even if there were few ways to design such a program, it does not seem likely that Dragon would have to use an identical structure and copy approximately fifteen out of fifteen commands." *Id.* at 966.

¹⁰ See *supra* note 7 for the text of section 102(b). *Baker v. Selden*, 101 U.S. 99 (1879), for example, which denied copyright protection for a new accounting system described in a book and which was partially codified by section 102(b), did not inquire into how many other systems of accounting existed. It was enough to state that a "system" of that type was not copyright subject matter, no matter how creative its author.

¹¹ 101 U.S. 99 (1879) (refusing to allow the stringent requirements of patent law to be circumvented by recognizing copyright in patent subject matter).

determine which elements should be protected by copyright and which should be either free for further incremental improvement by others or, if they meet the standards for patent or trade secret protection, be protected under those regimes.¹² Copyright law's broad "substantial similarity" test for infringement would unduly inhibit the development of improved software products if applied to nonliteral program elements like modular structure.¹³

This Comment argues that copyright should protect computer program code from verbatim copying or slavish mechanical or electronic translations. Other program elements, such as structure, sequence, and organization ("SSO") and elements of software interfaces, should not be protected by the copyright in the program code, that is, by the computer program copyright. If they are to receive intellectual property protection at all, that protection should be sought in patent or trade secret law or, in the case of nonfunctional elements of user interfaces (such as pictorial video game characters), in an independent copyright covering that element as a traditional copyright-protected work. Part I of this Comment discusses the facts of *Softel*. Part II articulates a test for separating protected "expression" from unprotected "idea" in a computer program, and

¹² See Julie E. Cohen, *Reverse Engineering and the Rise of Electronic Vigilantism: Intellectual Property Implications of "Lock-Out" Programs*, 68 S. CAL. L. REV. 1091, 1200 (1995) [hereinafter *Reverse Engineering*] (treating section 102(b) functionality as the touchstone for evaluating the copyright protection of program nonliteral elements).

¹³ *Whelan* supplies a good description of program structure in terms of its modules and subroutines:

The outline [of a solution to a complex programming problem] can take the form of a flowchart, which will break down the solution into a series of smaller units called "subroutines" or "modules," each of which deals with elements of the larger problem. . . . A program's efficiency depends in large part on the arrangements of its modules and subroutines; although two programs could produce the same result, one might be more efficient because of different internal arrangements of modules and subroutines. Because efficiency is a prime concern in computer programs (an efficient program being obviously more valuable than a comparatively inefficient one), the arrangement of modules and subroutines is a critical factor for any programmer. . . . As the program structure is refined, the programmer must make decisions about what data are needed [and related operational questions about data]. . . . [T]here are numerous ways the programmer can solve the data-organization problems she or he faces. Each solution may have particular characteristics—efficiencies or inefficiencies, conveniences or quirks—that differentiate it from other solutions and make the overall program more or less desirable.

Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc., 797 F.2d 1222, 1230 (3d Cir. 1986) (footnote and citation omitted). The term "modular structure" is thus the arrangement of a program's modules.

Part III applies that test to program SSO. Part IV then returns to the *Softel* decision to illustrate how and where the Second Circuit panel in that case lost its bearings and to speculate briefly on the implications of the *Softel* decision for the future.

I. FACTS OF *SOFTEL*

In *Softel*, the defendant Dragon produced interactive medical information programs. Dragon hired Softel to assist in developing some of the programming, especially code, that would retrieve and display video images produced by Dragon artists. After a personality dispute arose among representatives of the two parties, Dragon ceased using Softel's services but continued to use Softel's retrieval and display code in several new programs.¹⁴ The lower court found Dragon liable for these unauthorized uses of Softel code, and such finding was not at issue on appeal.¹⁵ However, following Dragon's subsequent development of new programs designed to work with different hardware and written in a different computer language, Softel asserted a second copyright infringement claim. This claim was that Dragon's new programs exhibited a substantially similar structure to Softel's programs in that they contained (1) a hierarchy of menus, (2) functional modules, (3) external files, and (4) English language commands.¹⁶ The district court concluded that none of these features could pass through the *Computer Associates* filters:¹⁷ hierarchical menus were "one of the most efficient and user-friendly interfaces," the use of functional modules and external

¹⁴ 118 F.3d at 958-61.

¹⁵ *Id.* at 960.

¹⁶ *Id.* at 966.

¹⁷ *Computer Associates* establishes the now widely followed three-step "Abstraction-Filtration-Comparison" test for determining substantial similarity of expression in computer program structure. 982 F.2d 693, 706 (2d Cir. 1992). The second step—filtration—is the key step for separating protected from unprotected elements of the structure. The idea is that all unprotected elements get filtered out of the substantial similarity analysis at this stage. The court describes the filtration process as follows:

This process [filtration] entails examining the structural components at each level of abstraction to determine whether their particular inclusion at that level was "idea" or was dictated by considerations of efficiency, so as to be necessarily incidental to that idea; required by factors external to the program itself; or taken from the public domain and hence is nonprotectable expression.

Id. at 707. Thus, there are at least three filters—efficiency, external factors, and public domain—and perhaps a fourth for "idea," although nothing in the decision tells us how to determine what the term "idea" means.

files was common in the industry, and the use of English language commands was logical and the most effective way for the programmer to keep track of such commands.¹⁸ Consequently, the district court denied infringement as to these programs¹⁹ and Softel appealed.

The Second Circuit vacated the district court's dismissal of these claims on the ground that the lower court did not properly apply the *Computer Associates* filtering step. Softel claimed that it had *combined* in an expressive way the four individual features filtered out by the lower court but that the lower court had ignored this argument and focused solely on the individual features.²⁰ Softel also argued that Dragon's programs employed fifteen commands that were "functionally identical" to the fifteen commands found in the Softel programs.²¹ The Second Circuit accepted this argument by analogy to traditional works of authorship, such as compilations of individually unprotected elements, abstract paintings composed of individually unprotected geometric forms, and literary works comprised of individually unprotected words.²² The court found that Softel had presented at least some evidence to support its argument that its way of combining the design elements was expressive, and that the lower court had either ignored or misanalyzed the argument.²³ Consequently, it remanded the

¹⁸ *Softel, Inc. v. Dragon Medical & Scientific Communications, Inc.*, No. 87 Civ. 0167, 1992 U.S. Dist. LEXIS 9502, at *63-*66 (S.D.N.Y. June 29, 1992).

¹⁹ The district court also made a factual finding that these programs "were not in any way derived from plaintiff's copyrighted work." *Id.* at *41. The court did not base any conclusions of law on this finding, however, and the Second Circuit did not discuss it. Interestingly, the Second Circuit *did* note that Softel had cross-examined Dragon's expert on whether the programs "evolved" from Softel's, as part of its discussion that Softel's argument was supported by some evidence. 118 F.3d at 966.

²⁰ See 118 F.3d at 963. Softel argued at trial, in language that was quoted by the Second Circuit, that it had created an "authoring language" that integrated the menus, touchscreens, modules, external files, and commands into original copyright-protectable expression. *Id.* at 965. Nowhere in the opinion is there a standard for what Softel meant by "expression," although implicitly Softel seems to have been claiming that "creativity" in the combination of features and absence of merger—the existence of other ways to accomplish the same result—means protectable expression. However, creativity, even coupled with the absence of merger, cannot in itself lead to a finding of protection when the section 102(b) exclusions are operative. See *infra* notes 39-41 & 65-68 and accompanying text.

²¹ See 118 F.3d at 965-66.

²² See *id.* at 964.

²³ See *id.* at 966.

case to the district court to determine whether the combination of these elements was expressive and, if so, whether Dragon had copied those expressive combinations.²⁴

II. "EXPRESSION" IN A COMPUTER PROGRAM

That copyright protects computer programs is no longer a matter for debate. The 1980 amendments to the Copyright Act, particularly the definition of a "computer program," make it clear that programs are copyright subject matter.²⁵ Equally clear is that the program copyright protects both source and object code from verbatim copying.²⁶ The major issue, which has never been explicitly addressed by Congress, is the scope of that protection. Because copyright protects only "expression," we must, therefore, determine what is meant by "expression" in a computer program in order to delineate the scope of protection.

²⁴ See *id.* at 967.

²⁵ In 1974, Congress established the National Commission on New Technological Uses of Copyright Works ("CONTU") for the purpose of advising Congress on how best to handle computer software under intellectual property law. In 1978, CONTU issued its report entitled FINAL REPORT OF THE NATIONAL COMMISSION ON NEW TECHNOLOGICAL USES OF COPYRIGHTED WORKS 3 (1978) [hereinafter *CONTU Report*]. Congress responded by adopting the *CONTU Report's* recommendations almost verbatim. The most important recommendation was to add to section 101 of the Copyright Act the definition of a computer program: "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101 (1994) (definition of "computer program"). It should be noted that the most natural reading of this statutory definition is that the computer program is the code, that is, the "set of statements or instructions" that bring about the program's intended function ("certain result"). While statutory language may be stretched when policy reasons seem compelling, in this instance it is perverse to deviate from a natural reading because this reading is fully congruent with the underlying policy of the statute. As this Comment explains, optimal intellectual property policy also demands that copyright protection in programs be limited to code.

²⁶ *Computer Assocs. Int'l, Inc. v. Altai*, 982 F.2d 693, 702 (2d Cir. 1992) ("It is now well settled that the literal elements of computer programs, i.e., their source and object codes, are the subject of copyright protection."). CONTU was also concerned with, and indeed only thought about, literal copying of code, such as photocopying source code or transcribing it from magnetic tape to disk. See *CONTU Report*, *supra* note 25, at 22. There is no reference in the *CONTU Report* to "nonliteral elements" of programs or to their creative "structure, sequence, and organization." See *A Coherent Theory*, *supra* note 6, at 67-68 & n.46.

A. "Expression" under the Computer Associates Test

The Second Circuit's decision in *Computer Associates* makes no attempt to define expression in a computer program, nor does it supply a standard for determining what is expressive. The closest *Computer Associates* comes to answering the question is in its filtering step, where it sets forth some of the things that are *not* expression, that is, aspects of programs that are not protected by the program copyright: elements dictated by considerations of efficiency (which, according to the court, makes them necessarily incidental to the unprotected idea being expressed) and elements dictated by external factors.²⁷ The court attempts to excuse its failure to define expression by resort to the hoary quote from Learned Hand (with reference to traditional literary works) that "nobody has ever been able to fix that boundary [between idea and expression], and nobody ever can."²⁸

Computer Associates' recognition that the program copyright does not protect efficient structures was a major step forward from the expansive interpretation of the idea/expression dichotomy for programs in the prior leading case, *Whelan, Inc. v. Jaslow Dental Laboratory, Inc.*²⁹ *Whelan* understood that efficiency was a prime concern in program design and that a program's structure (the arrangement of its modules and subroutines) was a large determinant of program efficiency.³⁰ It went on, however, to hold that the program copyright covered program SSO, even efficient SSO, provided there existed a variety of other structures (SSOs) that could do the basic job at hand (in that case, organizing a dental laboratory).³¹ Thus, under *Whelan*, the idea/expression dichotomy in a computer program was reduced to a question of merger. The result, however, is to allow intellectual property protection in *technology*—the traditional subject matter of patent law—without any showing by the rightholder that the technological innovation meets the stringent

²⁷ 982 F.2d at 707. The court here also restates the obvious point that elements taken from the public domain are not protectable expression.

²⁸ *Id.* at 704 (quoting *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930)).

²⁹ 797 F.2d 1222 (3d Cir. 1986).

³⁰ *Id.* at 1230; see *supra* note 13.

³¹ *Id.* at 1240.

requirements for patentability.³² There can be no serious doubt that new and nonobvious program SSO is eligible for patent protection.³³

Nevertheless, *Computer Associates'* failure to deal more directly with what might constitute "expression" at the higher levels of abstraction in a computer program³⁴ makes its application at those

³² This was one of the key grounds for *Baker v. Selden's* denial of copyright in an accounting system (a nonliteral element abstracted from the literal words describing it):

The novelty of the art or thing described or explained has nothing to do with the validity of the copyright. To give to the author of the book an exclusive property in the art described therein, when no examination of its novelty has ever been officially made, would be a surprise and a fraud upon the public. That is the province of letters-patent, not of copyright.

101 U.S. 99, 102 (1879). The Eleventh Circuit has recently expressed similar views about the importance of maintaining the distinction between patent and copyright:

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

Bateman v. Mnemonics, Inc., 79 F.3d 1532, 1541 n.21 (11th Cir. 1996).

³³ As the technology for designing computer programs, which in turn are the technology for operating computers, program SSO is patent subject matter and therefore patentable if it meets the other patent requirements. For example, the recently granted patent in the Lycos search engine covered a method for rapid digital searching on the internet; the claims described a series of steps implemented in the software, which is essentially the structure of its implementing program. See Michael Warnecke, *Breadth of Lycos Search Engine Patent Could Have Far-Reaching Implications*, 3 ELEC. COMMERCE & L. (BNA) 840 (July 1, 1998); see also *Relative Roles of Patent and Copyright*, *supra* note 6. This latter article argues that the 1996 Patent Guidelines, *Examination Guidelines for Computer-Related Inventions*, 61 FED. REG. 7478 (1996), which treat physical mediums on which computer programs are recorded as "articles of manufacture," have essentially mooted the patent subject matter inquiry for computer programs. The only patentability issues now are whether the applicant claims a new and nonobvious way of making computers work better.

This analysis is strongly supported by *State Street Bank & Trust Co. v. Signature Fin. Group*, No. 96-1327, 1998 U.S. App. LEXIS 16869, which says that practical utility or application is the sole factor that determines the section 101 subject matter question. If the claimed invention has practical utility, patentability is determined under sections 102, 103, and 112 of the Patent Act (novelty, nonobviousness, and disclosure). *Id.* at *19-20, *26. In another recent decision, a subject-matter challenge was not even raised. See *Reiffin v. Microsoft Corp.*, No. C-98-0266-VRW, 1998 U.S. Dist. LEXIS 10518 (N.D. Cal. July 10, 1998) (involving a patent on a technology called "multi-threading," which allows computers to switch from one task to another so rapidly that they seem to be performing both tasks at once). The defendant won a motion for summary judgment of invalidity on the ground that the claims omitted essential elements of the invention as originally disclosed in the written description. *Id.* at *25-26.

³⁴ *Computer Associates* recognized that source and object code were copyright protected. See 982 F.2d 693, 702 (2d Cir. 1992). Code, therefore, was found to be expres-

levels uncertain. This is clearly shown in the *Computer Associates* opinion itself. The court refers to a core of protectable expression remaining after the filtering process as a "golden nugget"³⁵ and hypothesizes a "quantitatively small misappropriation which is, in reality, a qualitatively vital aspect of the plaintiff's protectable expression."³⁶ The court seems not to have fully grasped that all programs are written to accomplish, as efficiently as possible within the constraints imposed by the hardware, other software, and economics, the function the program is intended to perform.³⁷ Program structure is not chosen whimsically, nor is it chosen for aesthetic reasons unrelated to function. A particular structure is chosen because the programmer believes that it will best optimize intended program operations under the given constraints. Within those constraints, the program should, for example, run faster, make fewer errors, crash less often, be less vulnerable to viruses, be easier to install and repair, and easier to learn. Where not everything can be the theoretical best (which is always the case), trade-offs must be made. In short, this is an *engineering* problem. Given these facts of technological life, it is impossible that any "golden nuggets" or even an important (let alone a qualitatively vital) aspect of the SSO could survive the filtering process.

B. "Expression" in Copyright Generally

The core problem at hand is that "expression" does not have any generally determinable meaning in copyright law, other than what the courts conclude to be protected in particular cases.³⁸ The

sive, since "expression" (and not idea, process, method of operation, etc.) is what copyright protects.

³⁵ 982 F.2d at 710.

³⁶ *Id.* at 714.

³⁷ Pamela Samuelson et al., *A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2353 (1994) ("At virtually all levels of abstraction above the literal text, a program consists of procedures, processes, systems, methods of operation, or their components."). They might have added that literal text (code) also consists fundamentally of a method of operation or procedure. Because literal text is copyright protected, we must assume that Congress intended at least that much of a limitation on the exclusions of section 102(b), which alone was a big change from traditional copyright. There is no evidence that Congress intended to make any further limitations on the fundamental exclusions from copyright protection set forth in section 102(b). See *A Coherent Theory*, *supra* note 6, at 67-68.

³⁸ Professor Cohen has argued that distinguishing between idea and expression within the realm of traditional works of art is fraught with judicial value judgments of artistic

terms "idea" and "expression" are not analytically useful words in copyright because they are basically conclusions for what the courts do, or do not, protect in specific circumstances. In fact, such terms can become downright misleading if one falls into the trap of attaching to them an intuitive, everyday meaning divorced from the underlying, complex copyright policies that seek to balance the public benefits of creation incentives against those of free dissemination of information and culture. It is tempting, for example, to label as "expressive" all those aspects of a work that result from the exercise of intellectual creativity, at least if there is a variety of ways to accomplish the same general result (absence of merger).³⁹ How-

merit. Amy B. Cohen, *Copyright Law and the Myth of Objectivity: The Idea-Expression Dichotomy and the Inevitability of Artistic Value Judgments*, 66 IND. L.J. 175 (1990). Perhaps Judge Learned Hand meant something similar when he stated, in the context of a traditional literary work, that no one would ever be able to fix the boundary between protected expression and unprotected idea. See *Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930); *supra* note 28 and accompanying text. Moreover, at least one commentator, writing before digital technology had reached a mass audience, argued for abandoning the idea/expression dichotomy altogether because of the impossibility of coherently separating one from the other. His conclusion was that we should explicitly recognize that copyright protects ideas. Robert Yale Libott, *Round the Prickly Pear: The Idea-Expression Fallacy in a Mass Communications World*, 14 U.C.L.A. L. REV. 735 (1967).

³⁹ *Feist Publications, Inc., v. Rural Tel. Serv. Co.* emphasizes the need for some minimal level of creativity in order for a work to qualify as an original work of authorship. 499 U.S. 340, 345, 346, 348, 362 (1991). This has led some courts to find protectable expression wherever they find creativity. A particularly egregious example of this is the Tenth Circuit's decision in *Autoskill, Inc. v. National Educ. Support Sys., Inc.*, 994 F.2d 1476, 1495 (10th Cir. 1993) (protecting a creative "keying procedure" or "system" used in a program to teach reading, without any recognition that both "systems" and "procedures" are excluded from copyright protection under section 102(b)); see also *Compaq Computer Corp. v. Procom Tech., Inc.*, 908 F. Supp. 1409, 1418 (S.D. Tex. 1995) (treating a five-item collection of numerical parameters used for predicting computer hard drive failure as a protectable compilation because it showed the requisite degree of creativity); *CMAX/Cleveland, Inc. v. UCR, Inc.*, 804 F. Supp. 337, 354 (M.D. Ga. 1992) (selection and arrangement of field names in a computer file structure equals expression); cf. *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 79 (D. Mass. 1990) (a pre-*Feist* case essentially equating expression with creativity: "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.").

At issue in *Feist* was a factual compilation, which is a work that expressly qualifies for copyright protection if it contains originality in its selection or arrangement of materials, subject to the traditional limitations on copyright such as merger. *Feist* does not hold the converse, namely, that any product of intellectual creativity is copyright protected. Nor does *Feist* hold that any creative selection or arrangement of materials is a protectable compilation. An automobile is a selection and arrangement of elements, such as a motor, radio, brakes, windows, sun roof, etc., and a wide variety of such selections

ever, if an element of a work is excluded from protection under the work's copyright by section 102(b) or the more general doctrine of *Baker v. Selden*,⁴⁰ such element will remain copyright unprotected no matter how much creativity went into its production or how many other ways there may be of accomplishing its result.⁴¹

Nonetheless, it is not true that everything is left to judicial value judgments. Explicitly or implicitly, the decisions are trying to

will still result in a vehicle that gets its occupants from point A to point B. A novel is a selection and arrangement of words, and a painting is a selection and arrangement of colors. However, the automobile is not copyright protected at all, and novels and paintings are protected as literary and PGS works, respectively, *not* as compilations. Cf. *Bibbero Sys., Inc. v. Colwell Sys., Inc.*, 893 F.2d 1104, 1108 n.3 (9th Cir. 1990) (rejecting the argument that a blank form could be a compilation because its inclusion within another category—that of blank forms denied protection by Copyright Office regulation—precluded its being copyright protected as a compilation). Traditional compilations, like automobiles, novels, and paintings, have what might be termed a “systematic organization,” but the various elements of a compilation, in contrast to these other works, are not used together as parts of an integrated whole. The content elements of compilations are related, in that they are usually of the same general type (e.g., telephone numbers, logarithms, judicial case reports), but users of compilations simply seek the individual items they need and extract those items without reference to any of the other contents at all.

⁴⁰ *Baker v. Selden* holds that patent subject matter cannot be the object of copyright protection:

To give to the author of the book an exclusive property in the art described therein, when no examination of its novelty has ever been officially made, would be a surprise and a fraud upon the public. That is the province of letters-patent, not of copyright. The claim to an invention or discovery of an art or manufacture must be subjected to the examination of the Patent Office before an exclusive right therein can be obtained; and it can only be secured by a patent from the government.

101 U.S. 99, 102 (1879).

⁴¹ Consider the creative design of a useful article that is inseparable from its utilitarian function within the meaning of section 101. 17 U.S.C. § 101 (1994) (definition of “pictorial, graphic, and sculptural works”). Presumably, no one would argue that such creativity results in copyright-protectable expression, because this statutory definition expressly excludes such inseparable utilitarian aspects of works from the PGS category. *Id.* The same reasoning applies to any element of a work excluded from copyright protection by section 102(b). See, e.g., *Lotus Dev. Corp. v. Borland Int'l, Inc.*, 49 F.3d 807, 816 (1st Cir. 1995) (expression is not copyright protected where it is part of a section 102(b) “method of operation”). It is true that the scope of protection in a computer program is not limited by the PGS separability test, because programs are literary rather than PGS works. It is also true that Congress has limited the exclusions of both section 102(b) and *Baker v. Selden* in application to program code, because code can be said to represent a process or method of operation, and yet, we know that Congress intended to protect it under copyright. See *A Coherent Theory*, *supra* note 6, at 60 n.19, 71, 100. There is no evidence, however, that Congress intended to protect noncode creativity under copyright, and there are good policy reasons for not extending the program copyright very far beyond code. *Id.* at 67-69.

balance—for the particular kind of work before the court—society's interest in recognizing and rewarding the author for her efforts and society's interest in having a free flow of information without the transaction costs of copyright that might hinder rather than promote further cultural development. As more and more copyright decisions are rendered, both bench and bar develop an increasingly finely tuned notion of how far protection should and does extend in particular kinds of works. While this fine-tuning remains imprecise for works within the traditional core of copyright—works of art, music, and literary fiction—the notion of a “thin” copyright for works like histories, biographies, technical works, scientific works, and rule books limits copyright protection to verbatim or near-verbatim language.⁴² Thus, while the plot of a novel or play might be “expression,”⁴³ the “plot” of a history is not.⁴⁴ If we are to decide the scope of copyright protection in a computer program by analogy to traditional works, rather than treating them as the *sui generis* functional works that they really are, the question is whether the most nearly apt analogy is to fictional novels and plays or to historical, scientific, and technical works. If the latter is the appropriate analogy, the scope-of-protection question essentially answers itself: The program copyright affords protection against literal copying and mere slavish translations of code but does not extend to higher elements of program SSO, regardless of how creative such elements may be.⁴⁵

⁴² E.g., *Landsberg v. Scrabble Crossword Game Players, Inc.*, 736 F.2d 485, 488 (9th Cir. 1984) (game strategy); *Miller v. Universal City Studios, Inc.*, 650 F.2d 1365, 1372 (5th Cir. 1981) (history); *Hoehling v. Universal City Studios, Inc.*, 618 F.2d 972, 980 (2d Cir. 1980) (historical theory); *Rosemont Enters., Inc. v. Random House, Inc.*, 366 F.2d 303 (2d Cir. 1966) (biography); *Continental Cas. Co. v. Beardsley*, 253 F.2d 702 (2d Cir. 1958) (legal form); cf. *Feist Pubs., Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 349 (1991) (copyright in a factual compilation is “thin”).

⁴³ *Sheldon v. Metro-Goldwyn Pictures Corp.*, 81 F.2d 49 (2d Cir. 1936) (Judge Hand finds that a film infringed the copyright in a play although both were based on a historical incident). But see *Nichols v. Universal Pictures Corp.*, 45 F.2d 119 (2d Cir. 1930) (Judge Hand denies infringement for two somewhat similar stories).

⁴⁴ *Hoehling*, 618 F.2d at 980 (historical theory on the cause of the Hindenburg disaster not infringed by a film adopting largely the same theory).

⁴⁵ If such elements really are creative, there is no reason that a patent would be denied should the creator seek one. See *supra* note 33.

C. A Policy-Based Test for "Expression" in Computer Programs

Another, and clearly better, way to give some content to the idea/expression distinction in the context of computer programs is to look to the policy underlying the decision to bring software under copyright protection. Computer programs are *sui generis* among copyright subject matter because of their inherent functionality. They are not like dictionaries or maps, which are useful only insofar as they supply information to human beings. A computer program is not intended to be "read" or "understood" by its target audience, let alone appeal to a user's sense of esthetics.⁴⁶ Rather, a program causes, through the *physical* flow of electrical currents, a complex set of switches inside a computer to set or reset in an orderly manner so that the *product* of program execution can be regarded as "information processing."⁴⁷

Thus, a computer program is quintessentially technology. As creative works of the human intellect, they are not distinguishable from other technological products, which receive intellectual property protection under patent and not copyright law. An executable program creates a special-purpose machine out of a general purpose computer. However, instead of designing that special-purpose machine with the methodologies of traditional electronic circuitry, technology has developed to permit the design by means of human-readable symbols (source code) that can be entered through a keyboard and translated into the binary electronic signals constituting the object code that governs operation of the computer. Therefore, computer programming is simply a new method of circuit design.

Consequently, the natural mode of intellectual property protection for computer programs should have been patent (or trade secret) law, as with any other advance in electronics. It is obvious, however, that computer program code, especially programs in object-code form, are vulnerable to fast, exact, and nearly costless

⁴⁶ Of course, what the program produces as output may seek to inform or appeal to aesthetic sensibilities. These output products may well constitute copyright subject matter. However, program interfaces in general, and user interfaces in particular, are not parts of the program. Rather they are products of the program—part of that "certain result" that execution of the program brings about. See *supra* note 25 (definition of "computer program"); see also *Computer Associates*, 982 F.2d 693, 703 (2d Cir. 1992) (screen displays "represent products of computer programs, rather than the programs themselves . . ."); *A Coherent Theory*, *supra* note 6, at 72-76.

⁴⁷ For a more elaborate description of computer programs as the technology for making computers work, see *New Protectionism*, *supra* note 1, at 36-41.

copying. Moreover, many computer programs fail to exhibit any significant advance in the methodology for making computers work better. Rather, they are simply straightforward, if sometimes costly, applications of well known principles and computer languages to a well defined problem. As such, they cannot meet the nonobviousness requirement of patent law and, without copyright protection or something like it, would be subject to slavish copying by competitors who neither make the same investment in development nor any improvements in the product. Many widely distributed programs would not be created or, if created, not widely offered, if they were not protected from this kind of misappropriative copying.

Therein lies the policy justification for protecting program code under copyright law, notwithstanding its technological nature. As stated at the outset, serious debate no longer exists over the protection of code by copyright law. The question that remains is whether the policy underlying the copyright protection of code applies to so-called "nonliteral elements" of the program and, in particular for the purposes of the *Softel* case, whether it also justifies the copyright protection of program SSO.

III. SSO AS "EXPRESSION" IN A COMPUTER PROGRAM

Before considering whether an anti-misappropriation policy supports the protection of SSO under copyright, inquiry should be made as to whether there are other social policies, such as reward for creative authorship, that might allow such a conclusion. It is certainly true, for example, that the copyright in literary fiction extends well beyond the verbatim language⁴⁸ and affords broad, exclusive rights to the author to control uses that cannot be characterized as slavish copying.⁴⁹ Many strong-protectionist commentators have used this as a basis for arguing that computer programs

⁴⁸ E.g., *Sheldon v. Metro-Goldwyn Pictures Corp.*, 81 F.2d 49 (2d Cir. 1936) (movie infringed rights in play); *Anderson v. Stallone*, 11 U.S.P.Q.2d 1161 (C.D. Cal. 1989) (entirely new script based on copyright-protected characters infringed).

⁴⁹ Professor Lemley has recommended that copyright law be changed to bring it more into line with patent law regarding the rights of authors who "improve" upon protected works. Mark A. Lemley, *The Economics of Improvement in Intellectual Property Law*, 75 TEX. L. REV. 989, 1084 (1997). If his argument is accepted, it would seem to end the debate in favor of non-copyright-protection of SSO independently of the argument presented herein.

should be protected broadly as well, including, in particular, protection of SSO.⁵⁰ These arguments, however, place the cart in front of the horse. The Supreme Court has consistently stated that reward to authors is an important but incidental benefit of our copyright system, the primary purpose of which is to promote the public interest in insuring broad availability of art, music, and literature.⁵¹ Congress undoubtedly has the power to change, within constitutional limits, some of the underlying policy, but Congress has never attempted to do so, either in general or with respect to program SSO in particular.⁵² Those claiming a policy basis for copyright protection of programs by treating them as if they were novels or plays or by assuming that the program copyright is aimed at rewarding technological creativity have the burden of showing that Congress intended such a deviation from accepted copyright policy. The only thing known for certain is that Congress, and its advisory committee CONTU, saw a need to protect code from fast, cheap, and error-free copying. Beyond that, Congress has simply not spoken to the issue.

More generally, there is good reason *not* to expand the copyright protection for computer programs much beyond code, and especially to SSO, unless SSO, like code, is vulnerable to incentive-eroding copying. Novel and nonobvious creative advances in SSO

⁵⁰ E.g., Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 HARV. L. REV. 977, 1008 (1993) ("computer programs are entitled to copyright protection under the same principles that govern other literary works.").

⁵¹ *Twentieth Century Music Corp. v. Aiken*, 422 U.S. 151, 156 (1975) ("The immediate effect of our copyright law is to secure a fair return for an 'author's' creative labor. But the ultimate aim is, by this incentive, to stimulate artistic creativity for the general public good."); see also *Fogerty v. Fantasy, Inc.*, 510 U.S. 517, 524 (1994) ("The primary objective of the Copyright Act is to encourage the production of original literary, artistic, and musical expression for the good of the public."); *Feist Publications, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 349-50 (1991) ("The primary objective of copyright is not to reward the labor of authors, but 'to promote the Progress of Science and useful Arts'"); *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417, 429 (1984) ("The monopoly privileges that Congress may authorize are neither unlimited nor primarily designed to provide a special private benefit. Rather, the limited grant is a means by which an important public purpose may be achieved."); *United States v. Paramount Pictures, Inc.*, 334 U.S. 131, 158 (1948) ("The copyright law, like the patent statutes, makes reward to the owner a secondary consideration."); *Fox Film Corp. v. Doyal*, 286 U.S. 123, 127 (1932) ("The sole interest of the United States and the primary object in conferring the [copyright] monopoly lie in the general benefits derived by the public from the labors of authors.").

⁵² See *supra* notes 25-26 and accompanying text; see also *A Coherent Theory*, *supra* note 6, at 66-72.

are patent subject matter.⁵³ Patent law is carefully designed to limit intellectual property rights in technological advances to those inventions that meet its more stringent conditions of protection, including approval of a trained examiner that the claimed invention meets patent law standards. Competitive copying, and often improvement, of unpatented inventions is an important part of the process by which technology improves. Thus, to recognize copyright protection, which arises automatically upon fixation with essentially no quality threshold, would allow creators in the particular field of computer technology to make an end-run around the limitations of patent law.⁵⁴

The above argument has been elaborated at length in my earlier works⁵⁵ and will not be reiterated here. The basic contention is that SSO, as opposed to program code, is no more vulnerable to incentive-eroding copying than most other technological products, which receive only patent or trade secret protection (and certainly not copyright protection). Anybody with a computer and a supply of blank disks can copy code, and in quantity, but reverse engineering a program to try to learn its structure is difficult and time-consuming, even for specialists.⁵⁶ Moreover, after some of the program's structural components are learned, the competitor must write noninfringing code, which must be tested, debugged, and marketed under a trade name different from the original. Even without copyright protection for SSO, this effort by the competitor takes time and money, which gives the original program creator important lead time in the marketplace and prevents the competitor from undercutting price too much when he finally markets his own noninfringing, functional product.⁵⁷ Consequently, there is no need for copyright

⁵³ See *supra* note 33.

⁵⁴ See *supra* note 32 and accompanying text.

⁵⁵ See, e.g., *supra* note 6.

⁵⁶ Andrew Johnson-Laird, *Software Reverse Engineering in the Real World*, 19 U. DAYTON L. REV. 843, 901 (1994).

⁵⁷ The *Whelan* court correctly observed that writing the code was only a relatively small part of producing a functional program:

By far the larger portion of the expense and difficulty in creating computer programs is attributable to the development of the structure and logic of the program, and to debugging, documentation and maintenance, rather than to the coding.

797 F.2d 1222, 1231 (3d Cir. 1986). The court concluded that a copyright on the structure and logic would protect the programmer's "most valuable efforts." *Id.* at 1237. However, it then rejected the defendant's argument that time and effort spent by the competitor who uses the original program's structure but writes all original code should

to protect SSO under an anti-misappropriation rationale and, given SSO's technological nature, there is no reason not to rely on patent and trade secret law—the traditional source of intellectual property rights in technological works—to achieve an optimal level of protection.⁵⁸

This general conclusion (that the program copyright should be essentially limited to code) and most of its analytical underpinnings receive additional support from Gillian C. Dempsey's recent, very thorough study of the economics of information.⁵⁹ Ms. Dempsey takes issue with the popular notion that essentially all technological information is readily appropriable by competitors as soon as a product embodying the information is released to the public. Rather than viewing information as a commodity that, absent intellectual property protection, would be unduly vulnerable to appropriation by competitors, she argues that it is often more realistic to treat information as a resource that grows incrementally through the

defeat a claim for infringement. The defendant's efforts, according to the court, were irrelevant to the infringement question, which asks "simply whether the copyright holder's expression has been copied." *Id.* This reasoning suffers from a fundamental midstream shift in the policy basis of the argument. It recognizes a copyright based on effort but refuses to consider effort in the infringement analysis.

Feist, of course, has now ruled out effort, or "sweat of the brow," as a source of copyright protection, but *Whelan* was decided before *Feist*. Although *Feist* later held that "sweat" copyrights in factual compilations ran counter to the statutory language, courts had been inclined to recognize them because failure to do so would result in a type of market failure: factual compilations are usually expensive to create but easy to copy once created. There was at least a policy justification for some of what these pre-*Feist* courts were doing, even if there was no basis in the statutory language. However, there is no basis in either law or policy to recognize a "sweat" copyright and then to find infringement when there is no market failure, that is, when the competitor has comparable costs and therefore cannot undercut price. If the SSO is truly valuable, its creator should go to the patent office.

⁵⁸ Professor Lunney has argued that the sole question in determining the copyright scope of protection should be whether copying affords a disproportionate or undue competitive advantage significantly greater than what we tolerate for non-copyright-protected works generally. Glynn S. Lunney, Jr., *Lotus v. Borland: Copyright and Computer Programs*, 70 TUL. L. REV. 2397, 2427, 2435-36 (1996) [hereinafter *Copyright and Computer Programs*]. While he would apply this approach to all copyright-protected works and not just computer software, applying this approach to software would seem to reach the same conclusion for program SSO as is presented herein. See also *New Protectionism*, *supra* note 1, at 87-88, 95 (we should prevent "copying activity that substantially eliminates the competitive commercial advantages enjoyed by technological innovators in other fields who are first in the marketplace with a new product.").

⁵⁹ Gillian C. Dempsey, *Knowledge and Innovation in Intellectual Property: The Case of Computer Program Copyright* (1998) (unpublished Ph.D. thesis, Australian National University) (on file with author).

interactions of many innovators.⁶⁰ Given the restrictions on these informational interactions imposed by patent and trade secret law, innovators usually have significant lead time in the market to secure a return on their investment in innovation, without any need for additional legal protection beyond the patent and trade secret regimes. Indeed, additional legal protection would reduce or eliminate the positive externalities that create, through such activities as reverse engineering, an ever-growing and widely distributed general pool of technological knowledge.

In particular, Ms. Dempsey distinguishes between information technology and a physical product ("information technology artefact") embodying that technology. She applies this conceptual framework to computer programs using a detailed consideration of their nature, including specific examples of compilation and decompilation. She argues that the distributed physical product (a program in object-code form) does not reveal the underlying technology (the source code). Competitors who do not copy code must engage in an expensive and time-consuming reverse engineering process, which eliminates the assumed easy appropriability of the commodity view of information. Ms. Dempsey ultimately concludes

⁶⁰ This is also the conclusion of a recent study inquiring into the reasons for the decline of Massachusetts's Route 128 as a leading center of innovation in electronics and the continued success of California's Silicon Valley. AnnaLee Saxenian, *REGIONAL ADVANTAGE: CULTURE AND COMPETITION IN SILICON VALLEY AND ROUTE 128* (1994) [hereinafter *REGIONAL ADVANTAGE*]. According to this study, Route 128 is dominated by a small number of independent firms that internalize most industrial activities and enforce practices of secrecy and corporate loyalty. This leads to hierarchical structures in which authority is centralized and information flows vertically within a single company, not horizontally from company to company. Silicon Valley's success, on the other hand, is attributed to its regional network-based industrial system that promotes collective learning:

The region's dense social networks and open labor markets encourage experimentation and entrepreneurship. Companies compete intensely while at the same time learning from one another about changing markets and technologies through informal communication and collaborative practices; and loosely linked team structures encourage horizontal communication among firm divisions and with outside suppliers and customers. The functional boundaries within firms are porous in a network system, as are the boundaries between firms themselves and between firms and local institutions such as trade associations and universities.

Id. at 2-3. I am grateful to Professor Leo Raskind for having brought Ms. Saxenian's work to my attention.

that while replication of code should be prohibited, other forms of imitative competition (provided that the competitors write their own source code) should be encouraged.

IV. *SOFTTEL* REVISITED

The Second Circuit panel in *Softel* neither noticed nor attempted to remedy *Computer Associates'* failure to articulate a coherent definition of "expression" at the nonliteral levels of a computer program. In fact, in terms of its technical copyright analysis, *Softel* represents a definite step backwards. *Softel* rotely applies copyright doctrine in generic terms, displaying no sensitivity to the technological nature of computer programs and no awareness that even traditional copyright exhibits wide variation in the scope-of-protection from one type of work to another.⁶¹ Thus, by failing to address the special nature of computer programs, the court mistakenly characterizes *Softel's* collection and arrangement of program modules—features dictated by technological considerations—as constituting elements of protectable expression in the form of a compilation.⁶²

Such characterization cannot be correct. Section 102(a) of the Copyright Act lists a variety of "works of authorship" that constitute copyright subject matter, including literary works, musical works, and PGS works. Section 103 extends the subject matter of copyright to compilations. The obvious intent of the drafters was to include factual compilations like lists that do not fit comfortably within any of the other categories.⁶³ There is utterly no indication that Congress intended a novel to be treated as a compilation (a collection of words) rather than as a literary work or that a painting should be treated as a selection and arrangement of figures or colors rather

⁶¹ See *supra* notes 42-44 and accompanying text.

⁶² See *supra* text accompanying note 22.

⁶³ Many traditional compilations fit within the formal definition of a "literary work," which is defined as a work "expressed in words, numbers, or other verbal or numerical symbols or indicia." 17 U.S.C. § 101 (1994) (definition of "literary works"). Perhaps a better explanation of section 103's explicit inclusion of compilations in copyright subject matter is that it was designed to insure that, for this particular class, judges would not deny copyright protection simply because the information contained in the compilation was factual. The definition of "compilation" in section 101 makes clear that the work is formed by the collection, coordination, and arrangement of the preexisting materials or data. 17 U.S.C. § 101 (definition of "compilation"); see *Feist*, 499 U.S. 340, 354-61 (1991).

than as a PGS work. Under the *Softel* court's analysis, nearly everything in the world is a compilation—a machine is a selection and arrangement of various mechanical parts, for example, and a new drug is a selection and arrangement of molecules. *Softel* wrote a computer program, which is a literary work under the Copyright Act. The issue is the scope of protection in that type of literary work. Emphasizing selection and arrangement of modules, and looking primarily for creativity (or the absence of merger) in that selection and arrangement, diverts the analytical focus from the underlying policies that should determine the appropriate scope of protection for this kind of work.⁶⁴

In particular, when dealing with functional subject matter other than program code, it is imperative that courts consider the exclusions from copyright protection commanded by section 102(b) of the Copyright Act.⁶⁵ *Baker v. Selden* illustrates that copyright protection does not follow merely from the exercise of intellectual creativity or from a determination that there are many ways to accomplish the same result (absence of merger). The accounting system in *Baker* was probably its most creative aspect, and yet copyright protection was denied to such system.⁶⁶ Moreover, while thousands of systems exist for keeping financial and accounting records, the absence of merger did not result in copyright protection.⁶⁷ Some highly creative products of the human intellect, such as the Theory of Relativity, are not protected by intellectual property law at all.⁶⁸ Others are protected by patent law, provided the strin-

⁶⁴ See *supra* note 39.

⁶⁵ The copyright in program code is not limited by section 102(b), because Congress intended to at least protect code under copyright; yet, code is clearly a "method of operation" or a "process." See *supra* note 41.

⁶⁶ *Copyright and Computer Programs*, *supra* note 58, at 2425.

⁶⁷ See *supra* text accompanying notes 8-12; *Reverse Engineering*, *supra* note 12, at 1147 ("[M]erger is a tool for identifying instances in which expression otherwise protectable by copyright is not protected. . . . The merger doctrine does not tell us how to identify those aspects of a copyrighted work that are ineligible for copyright protection in the first instance."); *A Coherent Theory*, *supra* note 6, at 81-82. The First Circuit's decision in *Lotus* compels this same conclusion. The trial court in *Lotus* had concluded that there were many ways to design a menu command hierarchy, but that fact was irrelevant because a menu command hierarchy was a copyright-unprotected method of operation. 49 F.3d 807, 812, 815 (1st Cir. 1995).

⁶⁸ Physical theories are not excluded from intellectual property protection because they are "laws of nature" rather than the creations of human intellect. Nor can they be excluded on the ground that they are "facts," notwithstanding Judge Easterbrook's recent statement that Einstein's famous mass-energy relationship $E=mc^2$ was an expression of fact. *American Dental Ass'n v. Delta Dental Plans Associated*, 126 F.3d 977, 979 (7th

gent conditions required for patent protection are satisfied. In both cases, these limits on the legal protection of intellectual creativity exist for good reason. The social cost of protection beyond these limits—in the form of fewer new products and reduced dissemination of knowledge—would exceed the benefit derived by society from the increase in incentives. Absent explicit congressional instructions, using copyright law to circumvent these well established limitations should not be allowed.

The *Softel* court appeared to be impressed that Dragon's program contained fifteen specific commands that were functionally identical to those in *Softel*'s protected program.⁶⁹ Applying merger analysis, the court did not find it likely that Dragon would have been required to use both the four structural elements in *Softel*'s program⁷⁰ and these same fifteen commands.⁷¹ However, the opinion never considers the potential impact of section 102(b) on the analysis. *Softel*'s program was designed to retrieve and display video images created by Dragon. The program evidently worked well, as the competence of *Softel*'s programmer was never questioned. It seems quite likely that the program would not have

Cir. 1997). While the relationship between mathematics and the physical universe remains the subject of profound inquiry, no one seriously believes that "nature" determines its ongoing course by solving the mathematical equations that human beings use to make (usually very approximate) estimations of nature's course and how we can change it. For over two centuries people thought Newton's laws of motion and gravity were "laws of nature." We now know that they were only approximations. No one knows for sure how long Einstein's refinements of Newton's laws, which manifest themselves only in extreme circumstances but reflect a radically different underlying philosophy, will remain what we call "laws of nature." All of these theories, and indeed all of mathematics, are products of human intellect. If the sole goal of copyright were to protect human intellectual creativity, protecting these theories would have to be very high on our list.

It is also true that a wide variety of physical theories may be, and indeed often are, available to explain natural phenomena. Many of these physical theories are discarded as additional evidence accumulates and as definitive experimental evidence rules out certain possibilities. There is, in fact, an infinite number of *incorrect* physical theories whose human expression would require highly creative thought. These theories are not "laws of nature," yet we do not protect them under copyright or any other branch of intellectual property law (except, perhaps, trade secret). Thus, neither intellectual creativity nor the mere absence of merger can be the basis for copyright protection. The threshold question must always be whether the subject matter is copyright protected. Given copyright subject matter (such as a computer program), the next question is whether particular elements of the work are unprotected under section 102(b) or other copyright exclusions, such as merger.

⁶⁹ See 118 F.3d 955, 965-66 (2d Cir. 1997), *cert. denied*, 118 S. Ct. 1300 (1998).

⁷⁰ See *supra* text accompanying notes 18-20.

⁷¹ See 118 F.3d at 966.

worked without all of these fifteen commands because it was designed it to do *this* job and not some other job. Presumably Softel's programmer did not put in extra commands that were wholly unnecessary. Moreover, these fifteen commands were sufficient to do the job, which is presumably why the Softel programmer did not use more than fifteen. Note that the claim was only that Dragon's commands were *functionally* equivalent to those of Softel. Dragon implemented these commands with its own code in a different programming language. Such commands were correctly filtered out of the infringement analysis by the lower court under *Computer Associates'* efficiency filter and by the First Circuit's reasoning in *Lotus v. Borland* that a hierarchical collection of functions constitutes a section 102(b) unprotected "method of operation" or "system."

Softel's bark may prove to be worse than its bite, at least in this specific case. Although its analysis harkens back to the discredited *Whelan* case, the actual error for which the court remanded was that the lower court had not even considered the application of *Computer Associates'* filters to the program SSO. The lower court on remand should, therefore, still filter functional elements aimed at efficient operation, which will include all of the SSO in question. Properly effected, such analysis should come to the result that Dragon is not liable for copyright infringement based on the similar SSO found in its independently coded programs.

The real problem with *Softel*, however, is its long-term effect on the analysis of other courts within and without the Second Circuit. Not only was the court's doctrinal analysis superficial and incorrect (in applying, for example, compilation concepts to novels, paintings, and computer programs); the court also failed to consider underlying policy, the fundamental limitations on copyright protection of section 102(b), and the need to separate patent and copyright subject matter to protect the integrity of the intellectual property system as a whole. Other courts are likely to be tempted to follow this doctrinally mechanical approach. To the extent they do, *Softel* represents a retrogression.

CONCLUSION

Courts that limit copyright protection in a computer program to code, as seems compelled by the statutory language⁷² as well as the intellectual property policies outlined above, need not fear that they are failing to follow the dictates of Congress or that they are leaving a valuable piece of United States technological leadership vulnerable to piracy.⁷³ Congress has simply not spoken to the question of the scope of copyright protection in a computer pro-

⁷² See *supra* note 25.

⁷³ Over 10 years ago I criticized the protectionist trend evidenced by cases like *Whelan* and surmised that the courts were basing decisions "on an implicit policy view that creativity and investment in program production need the incentive of a high level of protection." *New Protectionism*, *supra* note 1, at 35, 95. While *Computer Associates* and *Lotus* show a much deeper judicial understanding of both the law and its underlying policies, cases like *Softel* show that the task of bringing coherence to this subject is far from finished. Consider Judge Keeton's observation in *Paperback*: "It is no accident that the world's strongest software industry is found in the United States, rather than in some other jurisdiction which provides weaker protection for computer programs." *Lotus Dev. Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 75 (D. Mass. 1990). Judge Keeton's attribution of U.S. hegemony in the computer software industry to strong copyright protection is unpersuasive. United States software dominance actually began in the 1960s and 1970s when intellectual property protection for computer programs was thought to be either nonexistent or very weak. In particular, at least until 1986, everyone assumed that copyright would *not* cover the user interface—the work at issue in *Paperback*—because the only decision until that time had denied protection on functionality and section 102(b) grounds. See *Synercom Tech., Inc. v. Univ. Computing Co.*, 462 F. Supp. 1003 (N.D. Tex. 1978). The strongly protectionist decisions for user interfaces did not appear until 1986 and 1987. See *Broderbund Software, Inc. v. Unison World, Inc.*, 648 F. Supp. 1127 (N.D. Cal. 1986); *Digital Communications Assocs. v. Softklone Distrib. Corp.*, 659 F. Supp. 449 (N.D. Ga. 1987). This trend, of course, was short-lived as a result of the First Circuit's 1995 decision in *Lotus*. *Lotus Development Corp. v. Borland Int'l, Inc.*, 49 F.3d 807 (1st Cir. 1995), *aff'd by an equally divided Court*, 116 S. Ct. 804 (1996). It therefore seems unlikely that strong copyright protection could have been a major contributing factor to U.S. innovation in program interfaces.

This misconception results from looking only at the incentive side of the copyright equation, which assumes that more protection produces a greater incentive to create. One must not ignore the other side, which values free use to further technological development. See *supra* notes 59-60 and accompanying text. Indeed, AnnaLee Saxenian has persuasively argued that long-term success in high-tech innovation may actually require an environment in which ideas and know-how are actively exchanged, even among fiercely competitive firms. *REGIONAL ADVANTAGE*, *supra* note 60, at 35-36 (pointing out that job mobility in the Silicon Valley contributes to this process by establishing loyalties and friendships that transcend individual firms); see also Benjamin Kaplan, *AN UNHURRIED VIEW OF COPYRIGHT 2* (1967) ("Education, after all, proceeds from a kind of mimicry, and 'progress,' if it is not entirely an illusion, depends on generous indulgence of copying.").

gram.⁷⁴ Moreover, while it is important to stop piracy, it is equally important that intellectual property rules not hinder the complex processes by which technology develops. That implies application of the patent paradigm for protection of program elements that are not vulnerable to piracy in the way program code is. Those advocating copyright protection of SSO under a piracy rationale must show that SSO is, indeed, vulnerable to piracy. Unless that showing can be made, SSO should be treated like all other products of technology and left to its fate under patent or trade secret law.

Limiting copyright protection in computer programs to code and slavish translations of code has at least one incidental but important advantage, namely, a significant reduction in the complexities of obtaining proof in program infringement cases.⁷⁵ The scope of expert witness testimony would be reduced, and in many cases completely eliminated, because identity or near verbatim similarity of code is much easier to recognize than similarities of structure. It also removes an anomaly⁷⁶ that can arise from overly casual application of *Computer Associates'* efficiency filter: leaving efficient SSO unprotected but protecting *inefficient* SSO, leading to the perverse result that the plaintiff must try to prove that her program, which she is so eager to protect, is really not very good.⁷⁷ The copyright protection of computer programs, therefore, represents an example in which statutory language, underlying policy, and even incidental benefits come together to compel a fairly simple solution to an otherwise very complex, almost intractable, problem. That solution is to protect only the program code and slavish translations of code under the program copyright.

⁷⁴ See *supra* text accompanying note 52.

⁷⁵ See *New Protectionism*, *supra* note 1, at 55, 67.

⁷⁶ *New Protectionism*, *supra* note 1, at 83.

⁷⁷ *Reverse Engineering and Professor Miller*, *supra* note 6, at 1003-04.

