

SURVEY ON THE PATENT/COPYRIGHT INTERFACE FOR COMPUTER PROGRAMS

Pamela Samuelson*

I.	INTRODUCTION	258
II.	EXCLUSIVITY OR OVERLAP OF THE SUBJECT MATTERS OF COPYRIGHT AND UTILITY PATENT LAW IN GENERAL	260
	A. <i>Subject Matter Exclusivity in General</i>	261
	1. <i>Reasons Given for Exclusivity</i>	261
	2. <i>Reasons Given for Nonexclusivity</i>	263
	B. <i>When Did Patent and Copyright Cease to Become Exclusive (Or Were They Always Overlapping in Subject Matter)?</i> ...	264
	C. <i>Consequences of Subject Matter Overlap</i>	265
III.	ASSIGNING COPYRIGHT OR PATENT PROTECTION TO PARTICULAR ASPECTS OF COMPUTER PROGRAMS ..	266
	A. <i>Source and Object Code</i>	268
	B. <i>Structure of a Computer Program</i>	269
	1. <i>Instruction by Instruction Sequence of the Code</i>	269
	2. <i>Overall Structure or Design</i>	270
	3. <i>Intermediate Levels of Structure for Software</i>	270
	C. <i>Algorithms</i>	271
	D. <i>User Interfaces</i>	272
	E. <i>Specific Functionalities and Performance Characteristics of Software</i>	272
IV.	HOW COMPATIBILITY, EFFICIENCY, AND OTHER FACTORS AFFECT THE SCOPE OF COPYRIGHT	273
	A. <i>Compatibility Concerns</i>	274
	B. <i>Efficiency Concerns</i>	276
	C. <i>Standardization Concerns</i>	277
	D. <i>Reuse and Commonality Concerns</i>	278
	E. <i>Software Engineering Practices</i>	279
	F. <i>Market Factors</i>	280
V.	CONTINUING WITH PATENT AND COPYRIGHT OR SUI GENERIS PROTECTION FOR COMPUTER PROGRAMS	281
	A. <i>Sui Generis or Continued Patent and Copyright</i>	281
	B. <i>Proposals for Legislative Change</i>	283

*Visiting Professor of Law, Emory Law School; Professor of Law, University of Pittsburgh School of Law. The author has been serving as the co-chair (with John Sumner) of the Ad Hoc Subcommittee on the Patent/Copyright Interface for Computer Programs of the Computer Software Committee of the American Intellectual Property Law Association. She drafted the survey on which this article reports and performed the data analysis which is reported in this Article.

©1989 Pamela Samuelson

I. INTRODUCTION

The Computer Software Committee of the American Intellectual Property Law Association (AIPLA) formed an ad hoc subcommittee in the fall of 1988 to study the patent/copyright interface for computer programs. Because computer programs are both patentable and copyrightable,¹ difficult questions are arising about which aspects of computer programs can be protected by which law, and to what extent (if at all) the subject matters of the two laws overlap.² The AIPLA Ad Hoc Subcommittee seeks to aid in the clarification and resolution of some of these difficult questions.

In order to begin addressing these issues, the subcommittee chairs decided to survey the views of its members on a set of issues relevant to its charter. This Article reports the author's analysis of the results of this survey.³ The survey was intended as a device with which to inform the subcommittee members about the degree of consensus that existed among them about the degree of protection that is (or ought to be) accorded to computer programs by copyright and patent law, and the nature of the relationship between patent and copyright protection for computer programs.⁴

The survey had four basic parts. Accordingly, this report is divided into four sections, each reporting on the responding subcommittee

members' views about the set of issues set forth in that section of the survey. Part I will report on their views about whether or to what extent the subject matters of utility patent and copyright law are mutually exclusive, and the consequences they think flow from that exclusivity or overlap as it affects their understanding of the protection available under each law. Part II will report on the survey respondents' views about which aspects of computer software are or ought to be protectable by patent, copyright, both, either depending on some factor, or neither. The survey broke down computer software into component aspects (such as code, structure, and user interfaces) with instructions to say whether (or to what extent) each is or ought to be protectable under current intellectual property law. Part III will report on how the responding subcommittee members view such things as compatibility, efficiency, and standardization, as they affect the scope of copyright protection for computer programs. Part IV will report on the members' views about whether the existing copyright and patent systems adequately protect computer programs, or whether some legislative change would be desirable to improve the current schema for intellectual property protection of computer programs.

This survey report does not purport to be anything more than an informational device. In particular, it does not claim that the survey results it reports are the same as would have been found had a representative sampling been made of the views of intellectual property lawyers as a whole under scientific principles of sampling and statistical analysis. Although the survey sample size reported on here, like the size of the committee being sampled, is relatively small (26 survey responses), it is remarkable what diversity of views is reflected in even this sample, and the author believes that the survey results reported here are at least one useful datapoint (and at present, probably the best datapoint we have) from which to begin to understand the complex relationship between patent and copyright protection for computer programs.

In brief, the survey results show the following things: There is substantial consensus that there is a significant degree of exclusivity between patent and copyright law. Every one of the survey responses contained some statements indicating a significant differentiation in the roles of patent and copyright in the protection of computer programs. There is, however, a deep split within the respondent group on the issue of whether the two laws are completely mutually exclusive or overlap to some degree, as well as disagreement among those who think the laws overlap as to how much overlap exists.

As to the protectability of particular aspects of computer programs, the majority of the 26 survey respondents thought that source and object

1. Computer program-related inventions can be claimed as new "processes" or "machines" under 35 U.S.C. § 101 (1982). See *Diamond v. Diehr*, 450 U.S. 175 (1981). Computer programs were explicitly made part of the subject matter of copyright law in 1980. See Pub. L. No. 96-517, 94 Stat. 3007 (1980)(codified at 17 U.S.C. §§ 101, 117 (1982)).
2. Although there is now a considerable body of law review literature both on patent protection for computer programs and on copyright protection for computer programs, there has been, until this special issue of the AIPLA Journal, relatively little in the literature discussing patent and copyright protection in a coordinated manner. One notable exception is Maier, *Software Protection - Integrating Patent, Copyright, and Trade Secret Law*, 28 Idea 13 (1987). For a report on the views of those in the user interface field about the extent of patent and copyright protection that they think is appropriate, see Sameverson & Glushko, *Comparing the Views of Lawyers and User Interface Designers on the Software Copyright "Look and Feel" Lawsuits*, 30 Jurimetrics J. — (forthcoming 1989).
3. The statement of issues and survey form sent out to the subcommittee members can be found in Appendix A.
4. There were 44 members on the ad hoc subcommittee at the time of the first mailing of the survey to the subcommittee membership. Those subcommittee members who did not respond to the first mailing of the survey were sent a second copy of the survey along with a letter imploring their participation in the survey. Eventually twenty-six members of the subcommittee responded to the survey, and it is these twenty-six survey responses on which this report is based. After the author of this report had done her data analysis of these survey responses, she received some additional survey responses from a couple of people who were not serving on the committee, but were interested in the problems the survey addressed their completed survey answers. These respondents' views were consistent with the views of other subcommittee members. Because of their late arrival and the fact that they came from persons not on the committee, these respondents' views are not included in this report. However, it is worth noting that these additional responses were consistent with the results reported here on the survey.

code were protectable only by copyright, that functionalities were protectable only by patent, and that functional parts of user interfaces should be protected by patent and expressive parts by copyright. On the protection of structure, algorithms, and performance characteristics, there was no majority of opinion about which law (if any) protects which thing.

There was some consensus that compatibility and software engineering practices should have some effect on the scope of copyright protection for computer programs, somewhat less consensus about the effect of efficiency concerns, and on standardization, reuse, and market factors, respondents were about equally split on the issues.

The strongest consensus of all among the survey respondents was expressed as to their preference to continue to accommodate computer software within the existing patent and copyright systems, at least for the time being, although numerous suggestions were made about legislative action that would be desirable either to clarify some aspect of the law or to change current rules.

II. EXCLUSIVITY OR OVERLAP OF THE SUBJECT MATTERS OF COPYRIGHT AND UTILITY PATENT LAW IN GENERAL

Before asking the subcommittee members about their views on what aspects of computer programs were protectable by patent or copyright or neither or both, it seemed advisable to get their views on the general subject of whether utility patent and copyright law were mutually exclusive as to their subject matters.

If they thought there was some overlap between the subject matters of copyright and patent, they were asked a series of additional questions. One was whether the laws had ever been mutually exclusive, and if they had, when had this exclusivity ceased. Another was whether to the extent there was overlap in subject matters, one needed to elect as between them, or whether one could get both at the same time. Yet another was whether as to overlap subject matters, it was possible to get copyright protection for something that was patentable but failed for some reason (such as lack of invention) to qualify for patent protection, and then whether they perceived there to be any policy conflict if there was both copyright and patent protection for the same creation.

More than forty percent of the twenty-six survey respondents reported a belief that patent and copyright were completely mutually exclusive of each other in terms of their subject matters. Close to sixty percent of the survey respondents perceived there to be some degree of overlap as to these laws' subject matter. However, each and every one of

the survey responses contained some statement indicating a belief that there was some significant degree of exclusivity between the two laws as to what they might protect. As will be seen more clearly from the next section of this report, which will discuss the respondents' views about what aspects of software were protectable by each law, there was no consensus among the believers in subject matter overlap about how substantial the degree of overlap is.

A. Subject Matter Exclusivity in General

The first question in this section of the survey inquired about the subcommittee members' views as to whether the subject matters of utility patent and copyright in general were mutually exclusive, or whether they overlapped. Eleven of the twenty-six (43 percent) of the respondents said that they regarded the subject matters of patent and copyright law as completely mutually exclusive. None of them, however, interpreted this to mean that computer programs could be protected only by copyright or patent law. Rather, they regarded copyright and patent laws as having certain domains in which they could operate in the protection of computer programs, each law being the only applicable law within that domain.

Fifteen respondents to the survey (57 percent) thought that there was some degree of overlap between the subject matters of patent and copyright, yet they also perceived there to be some degree of exclusivity between the laws. Only a few perceived there to be an extensive overlap between the two laws, such that virtually all aspects of programs were protectable by both laws (so long as the requirements of originality, etc. were met), and even these persons still found some exclusivity. One of the subject matter overlap persons, for example, expressed the view that there was some exclusivity between these laws because "processes" were excluded from the reach of copyright under 17 U.S.C. § 102 (b), but included within patent's scope. Another subject matter overlap person said he favored letting the law evolve to define the overlap and its consequences.

1. Reasons Given for Exclusivity

Baker v. Selden,⁵ the exclusion of "process" from protection by copyright in section 102(b)⁶ and its inclusion in the subject matter of

5. 101 U.S. 99 (1879).

6. 17 U.S.C. § 102 (b) (1982). This provision states: "In 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."

patent law,⁷ and the "useful article" doctrine of copyright law (which excludes such articles from the subject matter of copyright)⁸ were the most frequently cited sources of support for regarding the subject matters of copyright and patent to be mutually exclusive. One commentator pointed to statements from *Baker v. Selden* indicating that subject matters protectable by utility patents are outside the scope of copyright. A second echoed that *Baker v. Selden* says that the "useful arts" are the subject matter of utility patent law, not copyright, and likened a computer program to a blueprint which copyright might protect, and likened its functional aspects to useful articles which are excluded from copyright but included in the patent domain. A third respondent regarded "utility" or "functionality" as the boundary line between the subject matters of patent and copyright law, as Congress has intended the laws to operate. Yet another said the exclusivity was clear from the subject matter provisions of each law.

One survey respondent quite eloquently expressed his view that exclusivity arose not just from the statute but from the Constitution⁹ as well:

"The distinction between utility patent and copyright protection is to be found, in my view, . . . in the constitutional and statutory distinctions between 'writings' of 'authors,' on the one hand, and 'inventors' and 'inventions,' on the other.

"The respective statutory terms are 'works of authorship,' for copyright, and 'process, machine, manufacture, or composition of matter,' for patents. In one of these categories — processes — the separation between patent and copyright is clear. Processes are expressly eligible for patent protection under 35 U.S.C. § 101; processes are expressly ineligible for copyright protection under 17 U.S.C. § 102(b).

"The remaining patent subject matters — invented or discovered machines, manufactures, or compositions of matter — differ from works of authorship, in my view, in the nature of the

creative processes which bring them into existence. Compositions of matter, for example, typically result from combining physical elements, as in chemical experiments. This is creative — inventive — activity, but has not traditionally been thought of as 'authorship.' The same kind of distinction can be drawn, I believe, between 'authorship' of 'works' and the 'invention' of 'machines' and 'manufactures.' Works of authorship are 'written' by 'authors;' patentable inventions are 'invented' or 'discovered' by 'inventors.'

...

"Courts have avoided overlap between utility patent and copyright protection in the past through appropriate interpretation of the nouns in the definition of patentable subject matter. For example, there is an old rule that 'printed matter' as such is not patentable. . . . According to Chisum, the courts have in general construed the term 'manufacture' in 35 U.S.C. § 101 in such a way as to exclude from it 'printed matter' that would constitute 'writings' in a copyright sense."

Several others who regarded the subject matters of patent and copyright to be mutually exclusive made more general statements to the effect that patent was available to protect much that copyright law regarded as ideas, and copyright was available to protect expressions of ideas. A few others commented that there had been some confusion about the issue recently as regards computer programs, but that the laws really were exclusive.

2. Reasons Given for Nonexclusivity

There was some diversity of opinion among those who regarded patent and copyright law as not being entirely mutually exclusive. One of these respondents observed: "The basic concepts of patents and copyrights were never exclusive but throughout the history of this country, they were implemented in an exclusive manner." Another said that the subject matters of the two laws may overlap to some degree, although usually being distinguishable. Several others regarded the subject matters of patent and copyright law as not entirely exclusive, but did not elaborate more on their answers. Still another said that the two laws are generally designed to protect different creative contributions, and in some areas they overlap.

Computer programs figured prominently in several other explanations of subject matter overlap. One survey respondent said he thought

7. 35 U.S.C. § 101 (1982) provides: "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title."

8. 17 U.S.C. § 101 (1982) (definition of "useful article"). The definition of "pictorial, graphic, and sculptural works," which is also found in § 101, refers to "useful articles" and makes them unprotectable by copyright. Many of the things which would be excluded from copyright under this provision are protectable by patent law as "machines" or "manufactures."

9. U.S. Constitution, Art. I, § 8, cl. 8 provides Congress with the power "to promote the progress of science and the useful arts, by securing for limited times to authors and inventors the exclusive right to their respective writings and discoveries."

patent and copyright had been exclusive until computer programs had come along. Another explained his view on subject matter overlap by observing that the Whelan decision¹⁰ had said that structure, sequence, and organization of computer programs was protectable by copyright, thereby extending the scope of copyright protection to that which patent was suited to protect, that is, to functionality and general concepts underlying a program. Still another gave computer programs and "pop-up" books as examples of subject matter overlaps between patent and copyright.

One survey respondent took pains to offer alternative definitions to the term "subject matter" in answering this part of the survey, saying that if "subject matter" meant "software," then no, he did not regard the laws as mutually exclusive, but if "subject matter" meant "process" (as in 35 U.S.C. § 101 and 17 U.S.C. § 102(b)) and "expression," then yes, they were exclusive. (Ordinarily, such a response would have been grouped with the exclusivity believers; but because this respondent answered "both, but it depends" in answer to the section applying patent and copyright to particular aspects of software, his response on exclusivity seems more appropriately characterized as one indicating that the two laws are not entirely exclusive.)

B. When Did Patent and Copyright Cease to Become Exclusive (Or Were They Always Overlapping in Subject Matter)?

The advent of computer programs was the most frequently cited explanation in answer to the question asked of the believers in overlap as to when patent and copyright had ceased to be exclusive. Five respondents pointed to computer programs as the development that signaled the end of exclusivity of the subject matters of patent and copyright law.

Two other respondents pointed to the *Yardley* decision¹¹ (which held that someone was not precluded from obtaining design patent protection for a design which had been copyrighted) as signaling the end of subject matter exclusivity for patent and copyright. Another commentator looked to the earlier Supreme Court decision in *Mazer v. Stein*¹² as the first sign of nonexclusivity, for it held that artistic aspects of useful articles might be protectable by copyright, without regard to whether they might be eligible for design patent protection.

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

11. *In re Yardley*, 493 F.2d 1389 (C.C.P.A. 1974).

12. 347 U.S. 201 (1954).

Two other survey respondents questioned whether *Yardley* and *Mazer v. Stein* were relevant to the issue of subject matter overlap for copyright and utility patent law, since both decisions involve design patents and copyright, where there is more ground for finding subject matter overlap because of the "ornamentality"¹³ requirement for design patent protection.

One other respondent identified the 1976 Copyright Act¹⁴ as the point at which the two laws had ceased to be exclusive. Some respondents indicated that the two laws had never been exclusive without making other comments. And one said that he wasn't sure and that it would take a study of what had been copyrighted and what had been patented to answer the question.

C. Consequences of Subject Matter Overlap

Only one of those who thought there was some degree of subject matter overlap for patent and copyright law favored the notion that someone would have to "elect" as between these forms of protection as to matters within the overlap area. One commentator said that the subject matter to be protected and the scope of protection desired dictated whether one should select one, the other or both. This same commentator pointed out that it was not necessary to "elect" copyright protection, for it arose automatically by operation of law. Three others said election wasn't necessary because the two laws offered different types of protection. One pointed out that neither patent nor copyright law required election, and if Congress had wished to require it, Congress could have put it in the statute.

Given the response to the election question, it is not surprising that a majority of the believers in subject matter overlap thought that copyright protection could be asserted to protect a feature that might be patentable subject matter, but that for some reason (such as lack of invention) did not qualify for patent protection. One commentator said that the existence of copyright protection depended on whether the thing was an original work of authorship, not upon whether patent protection might be available for it. Another respondent said that the more protection there was, the greater the incentives to be creative.

Three others said that the scope of protection offered by each law was different to explain their answer to this part of the survey. Two others

13. Compare 35 U.S.C. § 171 (1982) (design patent subject matter provision requiring an "ornamental design") and 35 U.S.C. § 101 (utility patent subject matter provision requiring "useful" qualities of a work).

14. 17 U.S.C. §§ 101, et seq. (1982).

differentiated between the functional parts which patent law might protect and expressive parts which copyright law might protect (seeming to signal some exclusivity in the roles for each law).

On the issue of whether there was any policy conflict if both copyright and patent protection was available for the same creative work, there was less agreement among the subject matter overlap people. Four of the overlap proponents thought there was some policy conflict presented by concurrent protection by copyright and patent. One of these four pointed to the different lengths of protection afforded by each law. A second said that there was a policy conflict in some instances, but not in all. A third said that in theory there was no policy conflict, but in practice there was a policy conflict because copyright had been given such an expansive interpretation that it had "sloshed over" into the patent realm. A fourth said that when copyright protection extends as far as in *Whelan*,¹⁵ there was a policy conflict, but that if the scope of copyright was properly limited, there would be no policy conflict.

Several others sought to explain why they thought there was no policy conflict by talking about the differences in scope of protection offered by each law, and on differences in the roles of patent and copyright in the protection of innovation. One said there was no policy conflict as long as the scope of protection granted by the courts reflects different creative contributions required to get each protection (*i.e.*, was original expression being protected, or a new and unobvious functional combination?). Another respondent explained that patent and copyright offered different types of protection to different aspects of the same work, and also said that he didn't think *Whelan* could be stretched to the point where the scope of copyright would be the same as patent in terms of protected subject matter. Yet another respondent said that courts in some copyright cases had, for equitable reasons, extended copyright protection to functional aspects of computer programs, without realizing that patent protection was available for functional aspects. This respondent believed that once the domain of patent law for software inventions was recognized, courts would see that extension of copyright to functional aspects of programs was not necessary.

One respondent thought that there was nothing wrong with overlapping protection under two independent laws, giving as an example copyright and trade secret law protection.

III. ASSIGNING COPYRIGHT OR PATENT PROTECTION TO PARTICULAR ASPECTS OF COMPUTER PROGRAMS

As an aid to understanding with more particularity how the AIPLA subcommittee members conceived of the proper domains of patent and

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

copyright law in the protection of computer software and the extent of overlap they perceived there to be for software, there was a section of the survey which asked what aspects of computer programs they thought were protectable by patent, copyright, both, or neither. An "it depends" response was also available as an answer.

There were nine items listed in this part of the survey. They were: 1) source or object code; 2) instruction-by-instruction sequence of the code; 3) overall design or structure of the code; 4) intermediate levels of structure of the software; 5) algorithms; 6) user interfaces; 7) specific functionalities of the software; 8) specific performance characteristics; and 9) an "other features" part, which allowed respondents to name other aspects of computer programs that they thought might be protectable by copyright, patent, and/or some other law or no law.

Interpreting the data from this part of the survey was somewhat difficult because it was not always easy to understand what respondents meant by their answers.¹⁶ But what struck the author most clearly from reviewing this part of the survey was that though there was some agreement among the survey respondents as to some matters, there was no consensus as to many important matters, and even where the bulk of the answers fell into one category or another, there is a wider diversity of views on important subjects than is desirable. Since the author believes that the software industry is in need of clear and certain answers to questions of the sort asked in this survey, and since intellectual property lawyers will understandably wish to be able to advise clients in accordance with how the law is generally understood, it is very disturbing that there are so many areas where there is as yet no significant consensus. These matters need to be resolved as soon as possible so that people (clients and lawyers) can start getting and receiving relatively consistent advice on software protection issues.

The issues as to which there was some agreement (*i.e.*, more than half of the twenty-six respondents agreeing on a point) were these: 1) that it was the proper role of copyright (but not patent) to protect source and object code; 2) that it was the proper role of patent law to protect specific functionalities of computer programs; and 3) that whether user interfaces were protectable by patent or copyright depended on what one was talking about, the artistic or expressive parts of user interfaces being in the domain of copyright, and the functional parts being in the domain of patent law.

16. For example, what was one to make of someone who described himself as a believer in complete subject matter exclusivity as between patent and copyright protection, and then in response to a question in this section of the survey circled "both" as an answer? Based on comments made in the explanation space, it was clear that an "it depends" more accurately reflected the person's view, and his answer was, accordingly, transferred to the "it depends" category.

On the protectability of algorithms and performance characteristics, a plurality of responses favored exclusive treatment by patent. But the protectability of structural aspects of programs was a subject on which the respondents were quite deeply divided.

Before getting further into the details of the answers given to this part of the survey, a few general comments on how the respondents handled this part of the survey are in order. All but two of the survey respondents did as requested and assigned to patent or copyright (or both or neither) the role of protecting particular features of programs. Many of these respondents offered some explanation of their views, and as they did so, they often expressed views seeming to reflect exclusivity concerns.

Only two respondents took a different approach in answering this section of the survey. One of these described his approach in this way: "I don't want to be tied down by abstract principles of what patent or copyright might be." This respondent answered "it depends" as his response to all nine items on the survey.¹⁷ Another respondent took the position that all that must be done to get both copyright and patent protection for any aspect of a program was to satisfy the statutory requirements of each law. The availability of protection, therefore, in his view, depended on satisfying the standards of each law, as to originality, nonobviousness, novelty, lack of merger of idea and expression, and the like.¹⁸

A. Source and Object Code

Copyright was identified as the appropriate form of intellectual property protection for computer program source and object code by a majority of the respondents to the survey. Only one respondent distinguished between source and object code, saying that source code was protectable only by copyright, but object code might be protected by both laws. Three other survey respondents also viewed both copyright and patent as appropriate forms of protection for the literal code of a program. But the weight of opinion of the survey respondents was for copyright only.

Several of the respondents who identified copyright as the appropriate form of protection for source and object code qualified their answers by

17. In answer to an earlier section of the survey, this respondent had indicated that he didn't think patent and copyright were entirely exclusive; yet he also thought that in some instances there might be a policy conflict if patent and copyright protection was recognized in the same creation.

18. Yet this respondent too, in the answer to the first section of the survey, had acknowledged that copyright did not protect those "processes" which patent might, again seeming to reflect a belief in some measure of exclusivity between the two laws.

pointing out that the literal code of a program, because it might be an implementation of a patented program-related invention (such as an algorithm), might infringe that patent if permission had not been obtained to implement it. This did not, however, affect their view that the code itself could only be copyright-protected. Two respondents pointed out that patent office policy seemed to preclude claims consisting of source code listings.

B. Structure of a Computer Program

In recognition of the fact that it is common in the computer programming field to think of programs as susceptible of analysis in terms of levels of abstraction, there was not just one question about the protectability of the structure of computer programs, but three questions on it. As programmers would view the matter, the lowest level in the hierarchy of abstractions of a program is the code. One level of abstraction above that might be the instruction-by-instruction sequence of the code. Up from that in the hierarchy of abstractions might be one or a number of intermediate structural abstractions of the program, and at some point below the apex of the hierarchy (which might be the general purpose or function of a program), there will generally be an overall design of the program. The survey, accordingly, broke the protectability of structure issue down into three issues: protection of the instruction-by-instruction sequence; protection of overall design; and protection of intermediate levels of structure.

Although, as indicated above, survey respondents were quite divided in their responses to this part of the survey, there was, nonetheless, somewhat more agreement on some issues than others. Before going into detail on each topic, it may be helpful to the reader to know that there was more support for the instruction-by-instruction sequence being the exclusive province of copyright (again subject to possible patent infringement liability if it was implementing a function for which a patent had been obtained) and more support for the overall design being the exclusive province of patent than for other positions. It was as to intermediate levels of structure that views diverged the most.

1. Instruction by Instruction Sequence of the Code

Half of the twenty-six survey respondents regarded copyright as the appropriate form of protection for the instruction-by-instruction sequence of the source code of a program. The other half seem to have been about evenly split between those who thought it depended and those who seemed content with the possibility of dual protection by both

laws. One of the respondents (who had professed himself a believer in the exclusivity of patent and copyright) indicated he thought that low level structure was probably mostly for copyright, but if pseudo-code, it might be patentable. Another said that if there was a functional reason for putting one instruction ahead of another, then that would be patentable; otherwise it would be copyrightable. But others merely commented that as long as the sequence represented a novel method, it might be patentable. Others who opted for "it depends" or "both" did not explain their answers. Most who assigned this kind of structure to copyright also did not explain their answers.

2. Overall Structure or Design

The split of opinion over which law should protect the overall design of software was deeper than the split over instruction-by-instruction sequence. Three of the respondents would assign the task to copyright. Eleven would assign it to patent. Very few of these fourteen respondents made any comment about their answers. One who did said that overall structure was not expression and not protectable by copyright, but it was functional, so it was patentable. Another who selected the patent only option remarked that patent would protect against functional equivalents.

Although there were a few respondents who answered that both copyright and patent might protect the overall design, more expressed an "it depends" opinion than a pure "both" position. One respondent, having first circled "patent" and then changing his mind and circling the "both" option made comments suggesting that he might more appropriately be regarded as an "it depends" person: "From a levels of abstraction standpoint, copyright should be available provided that it does not occupy territory reserved to works of function." Asking himself how to draw the line, he observed that methods were not protectable by copyright, but did not seem sure how to apply that to the protection of design issue.

One commentator expressed the view that a fanciful overall design for a piece of software might be copyrightable, but in general, the overall design would be functional and would be protectable only by patent. Another respondent thought that if highly original, the overall design of a program could be protectable by copyright, whereas if it was more conventional, copyright would only extend to the literal code.

3. Intermediate Levels of Structure for Software

The split of opinion among the survey respondents was deepest about the issue of what law should protect intermediate levels of structure

between overall design and instruction-by-instruction sequence. Five respondents answered copyright. One of these was the person who would use an originality test to decide how far up the hierarchy of abstractions copyright should reach, with conventional structures giving rise to no protection. Seven respondents answered patent.

Most of the remaining responses were of the "it depends" variety, with once again some soul searching going on about whether copyright protection for such structures might intrude into the realm of methods or functions which should not be protected by copyright. One respondent who selected the "both" option stated that if the functional design becomes detailed enough so that its implementation forms a sequence and structure, then that could manifest expression which copyright could protect. Another respondent said that intermediate levels of program structure were only protectable by copyright if fanciful. If functional, then only patent was appropriate.

C. Algorithms

Protection of computer program algorithms by patent law has been a hotly contested issue for a good many years.¹⁹ From reviewing the survey, it appears that the debate on this issue is not yet over, although the debate tends to center on whether algorithms are patentable or in the public domain. There were no survey respondents who thought that only copyright could protect algorithms, and very few who thought that both patent and copyright could protect algorithms.

The issue which divided the respondents about algorithms was much more over whether algorithms were protectable by patent, or whether they were protectable by neither patent nor copyright law. Six of the respondents thought that neither law protected algorithms. About twice that number thought that patent law could protect algorithms, although many of these qualified their answers by saying that only "non-mathematical" or "applied" algorithms were patentable. One respondent thought it would be a fair use to take an algorithm from a copyrighted work. Another stated that algorithms are "ideas" that are unprotectable by copyright. Two others observed that though copyright could not protect an algorithm, it could protect the expression of an algorithm; and one of these two limited his comment further by saying that copyright could only protect an algorithm's expression in code.

19. See *Gottschalk v. Benson*, 409 U.S. 63 (1972) and *Chisum, The Patentability of Algorithms*, 47 U. Pitt. L. Rev. 959 (1986).

D. User Interfaces

Five respondents to the survey thought that only copyright was the appropriate form of protection for software user interfaces. (One of these qualified his answer by saying that design patents might be available for some aspects of user interfaces.) Three others assigned user interfaces exclusively to patent law. But the majority of respondents were of the view that some aspects of user interfaces were protectable by copyright, and some by patent, but it depended on what aspect of a user interface one was talking about.

By far the most common response was that functional aspects of user interfaces were protectable by patent law, whereas expressive aspects were protectable by copyright. One respondent went into some detail, saying that if a functional advantage resulted from an aspect of a user interface (such as fewer keystrokes, or fewer data inputs, or the like), then it was patentable, whereas if the advantage flowing from the interface feature was more attractiveness or easier comprehensibility because of simpler sentences or the like, then it would be protectable by copyright.

E. Specific Functionalities and Performance Characteristics of Software

There was substantial consensus among the survey respondents that specific functionalities of computer programs were for patent law to protect. No one assigned the protection of functionalities to copyright, but three respondents took an "it depends" approach (without saying on what). Two respondents thought that functionalities were not protectable by either patent or copyright law, one of them saying that patent doesn't protect what something does, but how it does it.

There was less agreement, however, about which law protected specific performance characteristics of software. A plurality of opinion favored assigning such things to the patent system. One person, however, would have copyright serve this function. However, four respondents did not think either law should protect performance characteristics. One repeated his comment that patent law did not protect what something did, but how it did it. Another said this didn't rise to the level of protectable subject matter. A third said that copyright did not protect

specifications, but only code implementing specifications; he added that specifications were not an implementation so they weren't patentable either. A fourth said this was too general for protection.²⁰

Of the six respondents who thought either that both laws might apply to the protection of performance characteristics or that it depended, only one explained her position. She said that if the performance characteristic was functional, such as making the program faster or requiring less memory, then it would be patentable, whereas if it increased comprehension because of communication, then it could be protected by copyright. One other commented that this was a grey area.

IV. HOW COMPATIBILITY, EFFICIENCY, AND OTHER FACTORS AFFECT THE SCOPE OF COPYRIGHT

The author believes that the main reason that protection of computer programs by copyright law has been so troublesome for the courts is that copyright law has never protected a technology before.²¹ As a consequence, there are many questions about the application of copyright law to computer programs for which there are no clear answers, either in the statute or the caselaw, among which are these: 1) whether (short of exact duplication of the whole of a program) a firm has the right to develop similar code in order to be compatible with another firm's hardware or software, 2) whether similarities between programs that might be due to use of the same efficient design are infringing of the copyright, 3) whether technological goals, such as standardization or reuse of software abstractions, affect the extent of copyright protection for computer programs, 4) whether software engineering practices, such as "reverse engineering" of software, should affect the scope of copyright for software, and 5) whether market factors affecting the design of, for example,

20. The author has omitted discussion of other features of computer programs and of other issues, which the survey permitted the respondents to fill in. Among the "other features" which were identified as copyright-protectable were: manuals, screen displays, graphics, and videogames. Among the "other features" which were identified as patent-protectable were: software which transforms data into a new kind of output (e.g., speech processing heuristics for processing audio signals into digital templates), processes for producing graphics, and functional screens.

Among the "other issues" mentioned were these: ideas, scientific truths, and mathematical equations are in the public domain; scenes a faire are in the public domain; broad functional protection is not available to copyright owners; if an idea is unpatentable because it is obvious, then others are free to use it; patent is available for nearly everything about programs, and copyright should be limited; and trade secrets are also available to protect programs, but others have a right to reverse engineer.

21. See e.g., Samuelson, *Reflections on the State of American Software Copyright Law and the Perils of Teaching It*, 13 Colum. VLA J. Law & Arts 61 (1988).

software user interfaces should affect the scope of copyright protection.²² The AIPLA survey polled the ad hoc subcommittee members on these issues as well.

This part of the survey yielded less complete responses than other parts of the survey.²³ Four of those surveyed left the section entirely blank. (In the discussion below, these four nonresponsive surveys are not counted in the tallies on the issues.) Five others made only one general, across-the-board comment on the entire section. One respondent said simply "it depends" (without saying on what). Two others made a general statement in response to this section of the survey that they didn't see how these factors affected the scope of copyright. Yet another commentator said that he didn't think any of these factors should affect the scope of copyright except as they might relate to fair use and reverse engineering. And the fifth person to make an across-the-board response said that he would limit copyright protection to literal code (and minor variations on it), leaving to patent law the protection of the technological side of programs to which most of the factors related.

Even those respondents who made at least some individualized comments about one or more factors often left at least one blank space on this part of the survey form, so there is a more spotty pattern of responses to this portion of the survey than to its other parts. Still, all but one of the factors (reuse was the exception) evoked some response from a majority of the respondents, and many interesting observations were made concerning each factor.

A. Compatibility Concerns

Twenty people had some response to the question of whether or how "compatibility concerns" might affect the scope of copyright protection for computer programs. Seven respondents (about 35 percent) regarded compatibility as having no effect on the scope of copyright for computer programs. Three of these seven respondents said simply that compatibility had no effect on the scope of copyright. A fourth respondent went on to say that there should be no limit on the scope of copyright protection to achieve compatibility because it was always possible to

write a different program to achieve compatibility. Yet another likened cloning a computer program to translating a literary text to explain why he did not regard compatibility as a defense. A sixth respondent who did not regard compatibility as affecting copyright said that licensing could solve the problem. A seventh said that although a copyright on a computer program could block others from the market, that would ultimately have the effect of making the excluded firms more creative.

Thirteen respondents expressed some willingness to consider compatibility concerns in the determining the proper scope of copyright for computer programs. In addition to the two respondents who made general comments on the whole section, one saying "it depends" and the other raising the possibility of a fair use defense in some instances, there were a variety of reasons given by supporters for some degree of special treatment of compatibility in determining the proper scope of copyright for programs. Two respondents commented that, in general, the desire to make a compatible software product would not excuse copying of the whole or a substantial portion of the object code of a copyrighted program, but that, in general, copyright protection should not be construed so broadly as to preclude achieving functional compatibility with other software or with hardware systems. One of these commentators pointed out that it was sometimes necessary to structure programs similarly in order for them to interface properly with certain hardware, software, and perhaps even with users. Because it is one of the most useful attributes of software that it can be interconnected with other software and hardware, this respondent thought that copyright law should be interpreted so as to take advantage of the interconnectability of software systems, not to block their development.

Numerous others expressed similar sentiments. One said that compatibility was a functionality issue, and that copyright should not bear on functionality because the subject of compatibility is (or ought to be) outside the scope of copyright. Another said that courts shouldn't use copyright to prevent making a functionally compatible program. Two others said compatibility limits the scope of copyright, one adding that this was because compatibility was at the functional level. Another responded that the scope of copyright should be narrow enough so that someone can reverse engineer another firm's software to achieve compatibility. Another added that computer programs raise compatibility issues not shared by other types of intellectual or industrial property, and that public policy reasons should be brought to bear in considering compatibility as a fair use. Two other respondents regarded copyright protection for computer programs to be quite limited, one saying that

22. For a more scholarly view about the scope of copyright protection for computer programs, which includes a discussion of many of these same factors, see Report of Last Frontier Conference on Software Copyright Law, *— Jurim. J. —* (forthcoming 1989) (hereafter referred to as "ASU Report").

23. All sections of the survey except this one were completed by all respondents. The four who left this section of the survey blank effectively reduced the size of the sample to twenty-two, but even these twenty-two did not always answer all of the questions asked in this section. The reader must, therefore, here be more attentive to the size of the sample as to each question than as to other sections.

patent law precludes expansive protection for computer programs under copyright and the other regarding copyright as properly restricted to code only.

B. Efficiency Concerns

Efficiency was an issue on which the survey respondents were more divided than was the case with compatibility. Of the seventeen respondents who had some response to this factor, ten (about 60 percent) regarded efficiency concerns as potentially having some effect on the scope of copyright for computer programs; seven regarded it as having no effect.

Only three of these seven offered any reason for their assertion that efficiency should have no effect on the scope of copyright for programs. One said that it would not mean a loss of efficiency to require others make software in a different manner because that might result in a new creation of value. Another said that efficient software should be more protectable to reward the persons who came up with the more efficient way of doing it. (It should be noted here that this respondent was a supporter of a *sui generis* approach to computer program protection issues.) The third said he regarded performance advantages of software, such as efficiency, as having a bearing on the originality of the software; he regarded software which exhibited a higher degree of originality as more protectable than other software.

Of the ten who regarded efficiency as affecting the scope of copyright, five offered some reasons for their conclusions. One said that courts shouldn't force use of inefficient code sequences. Another said that copyright should not be interpreted so expansively as to subject the industry to the inefficiency of developing each new technology from scratch. A third said that a software developer should be able to use the most efficient method in developing a program unless it was patented. A fourth said that features of a program which relate to efficiency can be claimed as an invention, and under *Baker v. Selden*,²⁴ this made it unprotectable by copyright.

This author was the fifth to comment on efficiency as a factor affecting the scope of copyright. Once one gets into the realm of measuring relative efficiencies of programs, the author thinks one has gotten into the realm of functionalities, which is a realm in which copyright has no business. Although some might regard the "idea/expression merger

24. 101 U.S. 99 (1879).

doctrine"²⁵ as enough of safeguard against copyright being used to block reuse of the most efficient way of doing something in a program, the extreme difficulty of proving in litigation that the particular way the plaintiff implemented a function is the single most efficient way of doing it makes "idea/expression merger," in this author's view, a chimerical limit on the scope of copyright, rather than a real one. "Efficiency" should also be understood as a meaningful concept only in relation to specific goals of a programmer. The most efficient code by one parameter will not be the most efficient code by another parameter. Since all programmers aim to write efficient code, copyright should, in the author's view, not get caught up in protecting efficient aspects of programs.

C. Standardization Concerns

Standardization was an issue on which the respondents were about equally split. Nine respondents did not think that standardization had an effect on the scope of copyright protection for computer programs. Ten regarded it as potentially having some effect.

Four of the respondents who did not regard standardization as having an effect on copyright made some comments in explanation of their thinking on the issue. One pointed out that standard setting groups such as ANSI and IEEE had policies regarding standardization on patented technologies which could be brought to bear when standardization issues arose as to copyrighted software. Another respondent gave an example of a patented software technology that had been made an IEEE standard and licensed for a nominal fee. Still another said that if the industry wanted to standardize on copyrighted software, licenses could take care of the problem, but it should not affect the scope of copyright. A fourth noted that the value of a standard arose as much from acceptance in the marketplace as from technical merit, and that protecting intellectual property rights was the key to financing large scale acceptance of the product in the market.

Comments were varied among those who regarded standardization concerns as potentially having some effect on the scope of copyright. One said simply that standardization should have a limiting effect on the

25. It is standard copyright doctrine that copyright law protects "expression," but not "ideas," and that in the event that an idea and an expression are so inextricably interconnected that the two cannot be separated, that "idea" and "expression" are considered to be "merged," and copyright law will not protect expression when it has merged with idea in order not to give its author a monopoly on expression of that idea. See e.g., *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738 (9th Cir. 1971).

scope of copyright. Another said that the public interest in standardization should be considered in deciding whether the exclusive rights of copyright should be interpreted to attach to certain aspects of programs. Another commented that standardization was a functionality issue which was or ought to be outside the scope of copyright. Still another thought that copyright should not prevent the making of standards. One respondent thought that standardization was a waiver of proprietary rights. Two respondents regarded standardization as something that would normally not apply to things protectable by copyright, but only to things protectable by patent.

D. Reuse and Commonality Concerns

The section on reuse and commonality concerns was the least responded to section of the entire survey. In fact, only nine people had any response to it at all. This is probably because the terminology is unfamiliar to most of those who are not software engineers or at least followers of developments in the software engineering field. "Reuse" is a term that software engineers and computer scientists use to refer to an important goal that is thought to have the potential to dramatically increase the productivity with which software can be developed while also improving the maintainability, enhanceability, and reliability of software systems.²⁶ Although software engineers and theoreticians of software engineering may differ in their views about what kinds of reuse are likely to achieve the greatest productivity gains — reuse of actual code or reuse of designs and abstractions — they are in agreement that reuse is a promising initiative for the future.

The U.S. Department of Defense has been particularly active in supporting research and development involving software reuse. ("Commonality" is a synonym of "reuse," one which seems more favored in NASA circles.) Commonality and reuse are regarded as essential to the successful development of very complex systems, for without it, software systems may be unusably complex. Most of the technical work on software reuse issues has taken place without consideration being given to intellectual property rights.²⁷

Of the nine survey respondents who reacted to the reuse issue, five did not regard reuse or commonality as having any effect on the scope of copyright protection for computer programs, and four seemed to think it might have some effect on the scope of copyright.

26. See e.g., Report of the Defense Science Board Task Force on Military Software (1987).
27. Id. See also Samuelson, *The Need For Reform of the Software Licensing Policy of the Department of Defense*, 27 Jurim. J. 9, 44-45 (1986).

Only one of those who regarded reuse as having no effect on the scope of copyright had any comment on the issue. This respondent said that reuse of code should be copyright-protected.

Only two of those who regarded reuse as potentially having an effect on the scope of copyright made any comments about the issue. One likened reuse to the *scènes a faire* doctrine of copyright law, which treats standard plot devices for dramatic plays and novels as not protectable by copyright. The other (the author again) regarded reuse as an important technological initiative which should be encouraged by the law to the extent possible. Reuse goals, though, have to be balanced against the need for sufficient protection of software innovations so that there are adequate incentives to support software research and development efforts. Reuse of code might be licensed on an appropriate basis; reuse of abstractions ought, in the author's view, largely to be without intellectual property consequences. Properly understood, the boundaries of copyright law should not be implicated by software reuse so long as it is not code that is being reused. Reuse is part of any engineering discipline; software is just the latest of these disciplines to develop the need for reuse.

E. Software Engineering Practices

Of the fourteen survey respondents to react to software engineering practices as a factor potentially affecting the scope of copyright protection for software, four took the position it should have no effect, and ten seemed to think it should have some effect on the scope of copyright.

Two of the four who did not regard software engineering practices as affecting the scope of copyright made comments about the issue. One said that others should be able to build upon existing programs, but licensing was the way to deal with intellectual property interests. Another said that software engineering practices should adapt to be in accord with intellectual property law.

Of those who regarded software engineering practices as a potentially limiting factor, five mentioned reverse engineering as an example of a software engineering practice which should be legal under copyright law. One of these observed that the *E.F. Johnson*,²⁸ *Q-Co.*,²⁹ and *NEC v. Intel*³⁰ cases had recognized that although it might be necessary to make a copy of a computer program in order to reverse engineer it, as long as the code that was developed after the reverse engineering study was not

28. *E.F. Johnson Co. v. Uniden Corp.*, 623 F. Supp. 1485 (D. Minn. 1985).
29. *Q-Co. Industries, Inc. v. Hoffman*, 625 F. Supp. 608 (S.D. N.Y. 1985).
30. 10 USPO2d 1177 (N.D. Cal. 1989).

substantially similar to the code under study, no copyright infringement should be found.³¹ Another commentator observed that software engineering practices limit the scope of copyright, and tend to be at the idea level anyway.

One thoughtful comment (not made by the author) was this: "Present uncertainty in the scope of copyright protection based on current case law makes software engineering practices difficult, since software engineers are unable to tell what scope of protection a competitive work might be given. However, . . . if statutory limitations are followed in the future by courts, software engineering practices can become extremely predictable, with the only uncertainty being patents which have not yet issued or which are unknown to a developer who has decided to undertake a development program without doing a patent infringement investigation." This respondent believed that the software industry can adapt to the uncertainty brought about by patents in much the same manner as other industries have done.

F. Market Factors

"Market factors," like reuse, was a factor that seemed not to be well-understood by the survey respondents, for only thirteen responded to this question. The author of the survey had listed it as a factor potentially affecting copyright in large part because the Fifth Circuit Court of Appeals in the *Plains Cotton* case had referred to "market factors" as an explanation of why, despite the structural similarities between the defendants' and the plaintiff's cotton marketing software, there was no copyright infringement.³²

Only two of the respondents mentioned the *Plains Cotton* case and its cryptic reference to "market factors" as a factor affecting copyright. One of these two suggested that *Plains Cotton* should be understood as involving an unusual set of circumstances related to the cotton market, and should not be understood as creating a general "market factors" defense to software copyright cases. The other commentator (the author) thought that *Plains Cotton's* "market factors" analysis might have somewhat broader implications. When the design of software or software user interfaces is constrained by market factors, the author thinks that copyright protection ought to be limited in recognition of the possibility that two firms responding to the same constraints might independently develop similar software. The author perceives the *NEC*

31. See ASU Report, *supra* note 22, at ____.
32. *Plains Cotton Co-op. Ass'n v. Goodpasture Computer Service, Inc.*, 807 F.2d 1256 (5th Cir.), *cert. denied*, 108 S.Ct. 80 (1987).

v. Intel case³³ as another instance of market factors serving as a constraint on software design and as an explanation for noninfringing similarities between programs.

A more difficult market factors problem for copyright is that which arises when users have come to depend on a particular software system's interface and another firm argues that market factors arising from installed base and user training concerns justify using the same or a similar interface. Such a defense was established in the *Synercom* case³⁴; it appears to be one of the defenses in the *Lotus 1-2-3* spreadsheet case.³⁵

Of the seven survey respondents to regard market factors as having no effect on the scope of copyright, only four made any comment on the issue. Already mentioned was the *Plains Cotton* commentator. Another respondent said that market factors determined what people wanted to copy, not the scope of copyright protection. A third said that market success was an indicator of originality which ought to be protected. A fourth said only that software was a fast growing area of technology and should be encouraged.

Six respondents seemed to regard market factors as potentially affecting the scope of copyright protection, but only four made some comment about the issue. One was the author's commentary on *Plains Cotton*. Another said that compatible manufacturers stimulate competition. Another said that market factors related to functionality issues which were outside the scope of copyright. A fourth said that courts had been granting unduly broad patent-like protection under copyright, but wasn't sure what effect market factors should have on the scope of copyright.

V. CONTINUING WITH PATENT AND COPYRIGHT OR SUI GENERIS PROTECTION FOR COMPUTER PROGRAMS

A. Sui Generis or Continued Patent and Copyright

About 80 percent of the twenty-six respondents to the survey favored continuing to work within the copyright and patent systems to achieve the proper balance of intellectual property protection for computer programs, rather than adoption of a *sui generis* system of protection for programs.³⁶ A variety of reasons was given by respondents. One of the

33. *NEC Corp. v. Intel Corp.*, 10 USPQ2d 1177 (N.D. Cal. 1989).
34. *Synercom Technology, Inc. v. University Computing Co.*, 462 F. Supp. 1003 (N.D. Tex. 1978).
35. See Samuelson, *Why the Look and Feel of Software User Interfaces Should Not Be Protected by Copyright Law*, 32 Communications of the ACM 563 (1989).
36. Congress created a *sui generis* (of its own kind) intellectual property law for semiconductor chip designs in 1984. See e.g., Samuelson, *Creating a New Kind of Intellectual Property: Applying the Lessons of the Chip Law to Computer Programs*, 70 Minn. L. Rev. 471 (1985).

more frequently cited reasons was that copyright and patent laws had long histories of evolving to provide intellectual property protection to new types of innovations, and thus far at least, the respondents thought that it had not been demonstrated that patent and copyright could not adequately adapt to the protection of computer programs.

A couple of commentators reasoned that people had gotten used to using patent and copyright to protect software and that on the international scale, copyright, at least, had become something of a standard for the protection of computer programs, and thus radical changes, such as *sui generis* protection, would be quite disruptive.

One respondent thought that the current caselaw provided a strong foundation for resolving most disputes about software protection. Another expressed confidence that patent and copyright law were evolving toward more definite boundaries in the protection of computer programs. One commentator said he preferred to stick with copyright and patent law for computer programs because there were too many special intellectual property laws.

There was, however, some significant dissatisfaction with the state of the intellectual property law regarding computer programs, even among those who favored continuing with copyright and patent protection for them. Some commentators expressed concern about the lack of clarity about the scope of protection for computer programs. Some said that they regarded the current copyright and patent schemes, when properly interpreted, as striking the right balance, but were concerned that some judicial interpretations were not proper applications of existing law. Some others adopted a "wait and see" approach, some saying that they might favor *sui generis* legislation if some of the currently overexpansive interpretations of copyright continued. Some respondents raised questions about how copyright and patent would deal with such things as silicon compilers, automatic code generators, and expert system programs.

It is also worth noting that nearly a third of those favoring continuing the patent and copyright approach for protecting computer programs (*i.e.*, six of the twenty-one respondents) nonetheless favored some legislative action to change the law regarding computer program protection (of which more in section B).

The five respondents who favored a *sui generis* system of protection for computer programs also had a variety of reasons for their preference. One said that *sui generis* protection was needed because software intellectual property law was an amorphous area of the law and was in need of guidance. Another said that copyright principles were strained when applied to software, and that patents were too slow and expensive to get to be right for software, problems which a *sui generis* system could

address and resolve. Another favored a *sui generis* approach, but only as to the copyright (patent, in his view, was working just fine as is), arguing that the same reasons Congress had for adopting a *sui generis* approach for computer chips applied to computer programs. Still another said that neither copyright nor patent law was really constructed to suit the unique attributes of computer programs, and in trying to apply these laws, the courts ended up stretching both sets of laws, causing a variety of ad hoc problems.

The author of this report admits to being a fifth respondent to circle the "*sui generis*" entry on the survey form. She regards computer programs to be too much of a machine to fit comfortably within the copyright system and too much of a writing to fit comfortably in the patent system, and regards computer programs as stretching both systems to the point of transforming them into very different systems than they have been in the past. Even so, she recognizes that choices have been made, both in the U.S. and abroad, to work within the existing copyright and patent systems, and that it is, as a consequence, incumbent on those in the intellectual property community to work toward as successful accommodation of computer programs within the copyright and patent systems as can be accomplished.

B. Proposals for Legislative Change

The five respondents who favored a *sui generis* intellectual property law for the protection of computer programs, of necessity, favored legislative changes to address the issues that software raises for intellectual property law. One of the five tried to sketch out some features of a *sui generis* law for software. Another of the five, although favoring *sui generis* legislation, would limit the scope of copyright for programs to program code and videogame user interfaces.

Interestingly, six of the respondents who favored continuing with the patent and copyright approach (30 percent) nevertheless favored some legislative changes to improve the law relating to computer program protection. Five favored changes to copyright law, and one to patent law. The proposal to change patent law was for a clarification of the application of patent law to innovations implemented in software.

Those who favored changes to copyright law made the following proposals: One said that Congress should clarify the meaning of "expression" as it applied to computer programs. Another would have Congress require deposit of the complete text of source code as a condition of copyright registration for published computer programs, regarding the current Copyright Office policy for "secret copyrights" to be improper. Another respondent favored changing copyright law so that

artistic computer program screen displays could be registered for copyright protection, but not those resembling blank forms or containing only nonoriginal or limited expressive text. Two other respondents would have Congress explicitly make legal a privilege to make copies of computer programs for reverse engineering purposes. One of these two also recommended changes to copyright law to make it clear that copyright law does not protect the functionality of computer programs, to define the term "method," at least as regards computer programs, so that it would be clear what copyright did not reach under section 102(b),³⁷ and to shorten the duration of copyright protection for computer programs "on the basis that it's integral to a machine (computer) that defines the functionality of the machine."

This same respondent pointed out that difficult questions for intellectual property law as applied to computer programs are ahead. "Like mask works," he observed, "the form of software and its medium will always develop faster than laws designed specifically to deal with them. In the near future, silicon compilers will be able to design (automatically) hardware that embeds complex programs. In the near future, machines will be developed that automatically generate software. In the first instance, is hardware or software protected? In the second instance, who is the author?"

The author of this report concurs with this respondent that technological innovations of the sort he described will raise questions that are difficult for copyright law to address. At least as difficult are questions that information technologies, such as software, will raise for patent law. Much of the innovation in the computer software field is, and will in the foreseeable future be, in developing methods for the organization of information in ways that facilitate its effective use and retrieval. Much of the innovation, for example, presented by expert system programs is in the manner in which the knowledge base is organized and in the manner in which the heuristic rules which guide the operation of the analytic engine are engineered into the structure of the expert system program. Are these innovations copyrightable, patentable, both or neither?

Traditionally it has been copyright law that has addressed issues relating to the protectability of patterns of organizing information. Because computer programs are written for machine implementation, and designing computer programs is a process of designing a machine, it is understandable that patent lawyers will tend conceive of programs as machines and technological processes, even when the nature of the innovation being implemented is organizing information, something

37. 17 U.S.C. § 102 (b) (1982).

with which patent law has less experience. Perhaps the patent law will volve with ease to accommodate these kinds of innovations. Perhaps copyright law will instead, or in addition. Working now to understand and resolve the complex issues presented by the patent/copyright interface for computer programs may help to clarify the current state of confusion in the law and to anticipate future problems so that they can more easily be resolved.

Appendix

STATEMENT OF ISSUES FOR PATENT/COPYRIGHT INTERFACE SUBCOMMITTEE OF AIPLA

Although it is now settled that both patent and copyright protection may be available to protect some aspects of computer software, there is still considerable uncertainty about the proper scope of copyright protection for software, about the bounds of patent protection, and most importantly, what the proper relationship between patent and copyright protection is for software.

In the past there was little perceived need for an explicit definition of the relationship between patent law and copyright law because the subject matters to which each law applied were generally understood to be mutually exclusive. Because computer software is both a writing and a technology, both copyright and patent law apply to it. Because computer software is a very valuable and expensive intellectual property to develop, there is need for clear and meaningful protection of it, and intellectual property lawyers are using both patents and copyrights as means for protection. Hence, there is now a need for a definition of the relationship between patent and copyright protection for software.

One of the factors that has complicated the process of defining the relationship between patent and copyright protection for software is that at the time Congress passed the amendments to the copyright statute that formally admitted computer software to the copyright system, it appeared that patent protection would rarely be available to protect software. Copyright protection seems to have been adopted for software in part because of the absence of meaningful patent protection for software.

Because of the perceived absence of patent protection for software, Congress did not attempt to define the proper bounds of copyright protection for software in relation to the proper bounds of patent protection when it passed the computer software amendments to the copyright statute.

As a consequence, the early software copyright cases have tended to pay little or no attention to the availability of patent protection for software inventions in the process of trying to

Appendix (Continued)

identify the bounds of copyright protection for computer software. The most telling example of this tendency is the *Whelan Associates, Inc. v. Jaslow Dental Laboratories, Inc.* case, in which the Third Circuit Court of Appeals announced that everything about a computer program but its general purpose of function was protected "expression" under copyright law. (In deference to the copyright doctrine that holds that when "idea" and "expression" cannot be separated from one another and have merged, copyright law will not protect the expression in order not to give a monopoly on the idea, the Third Circuit made an exception to its general rule such that a feature of software that could be implemented in only one or a small number of ways would not be considered protected expression because the limited range of possible expressions caused idea and expression to merge.) The Third Circuit did not articulate a theory about the implications of the possible availability of patent protection for some aspects of software in developing its test for software copyright infringement. This has contributed to the theory held by some that the same aspects of software can be eligible for patent and copyright protection at the same time.

Though some think that the Third Circuit decision in the *Whelan* case went too far in carving out such an extensive scope of copyright protection for software, there is as yet no consensus about what more limited scope of copyright protection is appropriate, let alone about the patent/copyright interface for software. The *Whelan* case represents a lost opportunity for defining that relationship.

But circumstances have changed since adoption of the software copyright amendments. Patent law now provides meaningful protection for software innovation. Now that the courts have become more receptive to patents for software inventions and now that word is finally getting back to software firms and their lawyers that patent law can provide meaningful protection for computer software, it is time for serious thought to be given to the patent/copyright relationship.

The American Intellectual Property Law Association, with its rich expertise in intellectual property matter, is well suited

Appendix (Continued)

to make a significant contribution to resolution of this pressing dilemma of intellectual property protection. To that end, AIPLA's Computer Software Committee has formed a Subcommittee on the Patent/Copyright Interface for Computer Software. The Subcommittee co-chairs have posed the following questions to the Subcommittee's membership and invited member comments:

I. On the Relationship Between Copyright and Patent Protection in General

A. Do you regard the subject matters of patent (*i.e.*, utility patent) and copyright law in general to be exclusive?

Yes No Not entirely

Explanation:

B. If the answer to I-A is no, were patent and copyright law ever, in your view, exclusive as to their subject matters? And if so, when did they cease to be exclusive?

Ever exclusive?

Yes No Not entirely

When ceased being exclusive?

- after the *Mazer v. Stein* decision?
- after the *Yardley* decision?
- upon passage of the Copyright Act of 1976?
- upon passage of the computer software amendments to the copyright law?
- other (explain)

C. If in your view the subject matters of copyright and utility patent law now overlap, what are the implications of this overlap?

Appendix (Continued)

1. Does a creator have to make an election of protection as between copyright and patent law?

Yes No

Explanation:

2. Can a creator get both copyright and patent protection for the same creation (or same aspects of a creation)?

Yes No

Explanation:

3. Can a creator get copyright protection for the innovation if for some reason (*e.g.*, insufficient invention) a patent is not available?

Yes No

Explanation:

4. Do you think there is any policy conflict between patent and copyright protection for the same creative work?

Yes No

Explanation:

D. To the extent you do not feel that your answers to questions A through C give your full view of the proper relationship between patent and copyright law, please describe your view on this topic.

Appendix (Continued)

E. If your view on the proper relationship between copyright and patent law is affected by your view about the proper relationship between copyright and design patent law, please discuss your view on this relationship.

F. If your view on the proper relationship between copyright and patent law is affected by your view about the proper relationship between utility patent and design patent law, please discuss your view on this relationship.

II. Scope of Copyright and Patent Protection for Software

A. Please give your views about whether copyright or patent protection ought to be available for the following features of computer software:

1. Source or object code

Copyright	Patent	Both	Neither	It depends
-----------	--------	------	---------	------------

Explanation:

2. Instruction by instruction sequence of the code

Copyright	Patent	Both	Neither	It depends
-----------	--------	------	---------	------------

Explanation:

3. Overall design or structure of the code

Copyright	Patent	Both	Neither	It depends
-----------	--------	------	---------	------------

Explanation:

Appendix (Continued)

4. Intermediate levels of structure of software

Copyright	Patent	Both	Neither	It depends
-----------	--------	------	---------	------------

Explanation:

5. Algorithms

Copyright	Patent	Both	Neither	It depends
-----------	--------	------	---------	------------

Explanation:

6. User Interfaces

Copyright	Patent	Both	Neither	It depends
-----------	--------	------	---------	------------

Explanation:

7. Specific functionalities of software

Copyright	Patent	Both	Neither	It depends
-----------	--------	------	---------	------------

Explanation:

8. Specific performance characteristics of software

Copyright	Patent	Both	Neither	It depends
-----------	--------	------	---------	------------

Explanation:

9. Other features of software:

- protectable by copyright?

- protectable by patent?

- protectable by both?

Appendix (Continued)

- in the public domain?
- protectable by some other law?

B. Under traditional principles of copyright and patent law, how do the following considerations affect the scope of copyright protection for software?

1. compatibility concerns?
2. efficiency concerns?
3. standardization concerns?
4. reuse and commonality concerns?
5. software engineering practices?
6. market factors?

III. *Sui Generis* Law for Software or Accommodation of Software into Existing Copyright and Patent Systems

A. What in your view would be preferable: to adopt a new *sui generis* law for computer software or to continue to accommodate computer software into the existing patent and copyright systems?

Sui generis

Continued patent/copyright

Explanation:

Appendix (Continued)

B. What if any statutory changes do you think need to be made to either copyright or patent law to make the accommodation of computer software into these laws more successful?

- no changes needed
- the following changes needed to copyright law
- the following changes needed to patent law