Chapter 1

Summary, Issues, and Options

Contents

Boxes

Chapter 1
Summary, Issues, and Options

Summary

Background and Study Approach

When a new technology impinges upon the intellectual property system, questions arise concerning both the scope and appropriateness of its protection and the effectiveness of the system in promoting "the Progress of Science and useful Arts. '[1] A series of OTA reports has explored the intellectual property challenges presented by new information technologies—particularly by the move to electronic representation of information and the proliferation of digital means of transmission, adaptation, and copying-and by biotechnology. Looking across these challenges, it would appear that, although the technological, economic, social, and industrial particulars differ, many of the tensions and issues that arise are general, rather than technology-specific.

In 1984, the House and Senate Committees on the Judiciary requested that OTA examine the impact of recent and future advances in communication and information technologies on the intellectual property system. In response, OTA prepared the 1986 report, *Intellectual Property Rights in an Age of Electronics and Information.*[2] That report found that technological changes in information technologies offer opportunities for social and private gain, but at the same time challenge existing business practices and legal doctrines. It examined the impact of new technologies on the effectiveness of intellectual property rights, including the right to control reproduction of copyrighted works, the right to control publication and performance of works, and the right to control the making of derivative works.[3] The report identified three types of information-based products—work *of art, works offact, and works of function-and* concluded that basic differences among these types of works create difficulties for the current intellectual property regimes.[4]

A 1989 OTA report, New *Developments in Biotechnology: Patenting Life-Special Report,*[5] examined challenges that biotechnology-specifically, the patenting of nonhuman living organisms—poses for the patent system (see box 2-A in ch. 2). Many of the questions and difficulties identified in that report (e.g., questions concerning the appropriateness and scope of patent protection, the newness of the technology, institutional difficulties in establishing a repository of prior art and in administering patent prosecution) have also been noted in this study[6] Thus, it appears that-although the particulars differ—many current questions concerning the

[1] U.S. Constitution, art. I, sec. 8.

[2] U.S. Congress, Office of Technology Assessment, *Intellectual Property Rights in an Age of Electronics and Information,* OTA-CIT-302 (Melbourne, FL: Kreiger Publishing Co., April 1986).

[3] For example, technologies that lower the cost and time required to copy, transfer, or manipulate information and intellectual property can make works more accessible, make them more valuable to consumers, and make using them more convenient. But, these technologies can also make enforcing intellectual property rights more difficult, potentially reducing financial incentives to produce new works. For more discussion of technological change and the enforcement of intellectual property rights, including impacts on print, music, video, and other media, see *Intellectual Property Rights inan Age of Electronics and Information,* op. cit., footnote *2,* pp. 97-123.

[4] *Intellectual Property Rights in an Age of Electronics and Information,* op. cit., footnote 2, pp. 64-68 et seq. Among other findings, the 1986 report identified particular difficulties with respect to achieving the public-policy objectives of our intellectual property system when applying it to computer programs (as works of function) and computer database systems (at the intersection of the factual and functional domains). (Ibid., pp. 67 and 78-88.) These included questions regarding the appropriate scope of copyright protection for programs (e.g., how "expression" is interpreted), patent protection for computer processes, and reverse engineering.

[5] U.S. Congress, Office of Technology Assessment, *New Developments in Biotechnology: Patenting Life—Special Report,* OTA-BA-370 (Washington DC: U.S. Government Printing Office, April 1989).

[6] In many respects, OTA has observed that the difficulties associated with the accommodation of software-related inventions within the Patent regime are similar to those experienced in biotechnology. There are some historical differences, though. For biotechnology, the industry had been in general agreement concerning the desirability of patent protection and the major controversy within the industry was over the scope of such patents. (See footnote 5.) In software, OTA has observed that (in addition to concerns about scope) software developers' opinions are somewhat divided concerning the general desirability of patents for software-related inventions.

More recently, however, controversy over biotechnology patents has renewed, this time concerning the subject matter, as well as scope, of patents for gene sequences. This controversy (in industry, academia, and government) was prompted by the National Institutes of Health's announcement that it planned to file patent applications for thousands of complementary DNA sequences, even those whose function is unknown. See Leslie Roberts, "Genome Patent Fight Erupts," *Science,* vol. 254, Oct. 11, 1991, pp. 184-186.

availability, scope, and administration of patents for computer processes and algorithms are not unique to software technology; rather., they are general questions that may arise when any new technology is introduced into the patent system.[7]

Another 1989 OTA report, *Copyright and Home Copying: Technology Challenges the Law,*[9] examined noncommercial, private use of copyrighted works and the implications of digital media and recording technologies, particularly for home audio recording. That report found that intellectual property laws serve to define the boundaries between *permissible* and *prohibited uses* of works; technology, driven by the social and economic objectives of its users, defines the frontiers of *possible* uses and feasible enforcement of boundaries.[8] OTA found that technological changes and trends[10] that substantially alter the nature and extent of possible uses, or the feasibility of enforcing prohibitions against certain uses, give rise to tensions between users and proprietors and may make modification or clarification of the law desirable.[11] In some cases, OTA found, new technologies (e.g., copy protection) may have the effect of a de facto change in the law.

In early 1989, the Subcommittee on Courts, Intellectual Property, and the Administration of Justice of the House Committee on the Judiciary (now the Subcommittee on Intellectual Property and Judicial Administration) requested OTA'S assistance in considering the issues related to the protection of computer software in a changing international marketplace. OTA prepared a background paper, *Computer Software and Intellectual Property,*[12] that examined current application of copyright, patent, and trade secret laws to computer software. It also provided an overview of the often conflicting views and concerns of various stakeholders among industry, academia, and the public at large.

Many of the tensions concerning intellectual property protection for a new technology are not necessarily technology specific.

Current Study Approach

In its prior work, OTA had identified several factors that contribute to the complexity of the software debate: 1) the nature of software technology itself, which makes it difficult to fit software into the current framework of copyright and patent law; 2) the rapid pace of technological change in computer hardware and software; 3) difficulties in reconciling cultural and definitional differences between the legal and technical communities; and 4) complications arising from the international scope of software markets and technologies. 13 In planning

[7] However, software is presently distinct in that multiple modes of legal protection can apply simultaneously to portions of a manifestation of the technology+. g., copyright for a program containing trade secrets, patentable software processes contained in a copyrighted program, etc.

Questions of patent scope and administration are not new—for a discussion of similar concerns in the 19th and early 20th centuries, see Fritz Machlup, *An Economic Review of the Patent System,* Study of the Subcommittee on Patents, Trademarks, and Copyrights of the Committee on the Judiciary, United States Senate, Study No. 15 Pursuant to S. Res. 236, 85th Congress, 2d Sess. (Washington, DC: U.S. Government Printing Office, 1958), ch. IV.

[8] U.S. Congress, Office of Technology Assessment, *Copyright and Home Copying: Technology* Challenges *the Law*, OTA-CIT-422, (Washington DC: U.S. Government Printing Office, October 1989).

[9] These legal bounds are based on the Constitutional intellectual property bargain, tempered by the feasibility and efficiency of enforcement. See *Copyright and Home Copying: Technology Challenges the Law,* op. cit., footnote *8,* ch. 2 and ch. 3.

[10] The technological trends identified in the 1989 report were: 1) the movement to *digital representations* of music, video, and Other types of entertainment and information available to consumers; 2) the *erosion of niche boundaries* used to categorize copyrightable works according to their content (e.g., audio, video, computer software) or physical format (e.g., audiotape, videotape, computer disc); 3) the *emergence of new delivery infrastructures* to deliver information and enrtainment (e.g., fiberoptic cable, interactive cable services); and 4) the efforts of some copyright proprietors (e.g., in sound recordings and motion pictures) to develop and implement *technical means for copy protection.*

[11] Similar technological trends are relevant *to* the present study, especially the blurring of niche boundaries and the emergence of new infrastructures for delivering computation and for interacting with software and hardware. When stored or executed in a machine, software and data are in "digital" representations and thus can be interacted with, copied, or manipulated easily and efficiently. At the same time that new delivery infrastructures such as high-speed networks are being deployed, important new technologies like hypermedia, virtual reality, and scientific visualization are blurring content-based niche boundaries.

[12] U.S. Congress, Office of Technology Assessment, *Computer Software and Intellectual Property,* OTA-BP-CIT-61 *(New* York, NY: Stockton Press, 1990). The background paper was released at the subcommittee's second day of oversight hearings (Mar. 7, 1990) on the topic of computers and intellectual property, at which OTA presented testimony. OTA had previously submitted a staff paper, ''Intellectual-Property Protection for Computer Software,'' to the subcommittee to assist in preparations for the first day of oversight hearings on that topic (Nov. 9, 1989).

[13] See Computer *Software and Intellectual Property,* op. cit., footnote 12, ch. 1.

and carrying out this study, OTA’S objectives were to:

- understand the characteristics of software as a technology, as well as identify the relevant technological changes and trends that will confront Congress;
- explore the relationships between the legal protection of software and incentives for innovation;
- understand the market, trade and policy implications of the emerging global software indusV ;
- identify current intellectual-property challenges presented by software and computing technologies and anticipate future challenges from technological developments in computer software, operations, and architectures.

In the course of this study, OTA sought out the opinions, positions, attitudes and perceptions of the stakeholders in computer software protection, including individuals from academia and the research communities, the legal profession, the computer software and hardware industries, government agencies, and the public at large. This was accomplished through personal interviews and correspondence, and through public participation in the study’s advisory panel and workshops. Each of the workshops focused on a specific set of issues or perspectives:

- Software Engineering Technology and Intellectual Property Issues (Sept. 24, 1990);
- Software-Developer Issues (Sept. 25, 1990);
- Public-Interest Issues (Dec. 7, 1990);
- Digital Libraries, Electronic Publishing, and Intellectual Property (Feb. 11, 1991);
- User Interface Technologies and Intellectual Property (Apr. 18, 1991);
- Patent, Copyright, and Trade Secret Protection for Computer Software (June 20, 1991).

In each of these workshops, as in the overall study, OTA sought to explore the dimensions of the software debate by examining software technology and its distinctive characteristics, asking questions such as:

- *What are the characteristics of the technology? How does it advance?*
- *What aspects of the technology are most important to society? To a proprietor?*
- *What might a proprietor want to secure rights for? Why? What are the private and public ramifications of granting or not granting these?*
- *How do these private and public objectives relate to current law? Do existing legal modes provide appropriate protection? Can they be implemented effectively and efficiently? If not, what might be done?*

This sequence of examination was not always possible. The published literature and the usual terms of the software debate tend to focus *on positive analyses*[14] *of Current law and* case law (e.g., whether copyright encompasses program command structures, whether certain computer algorithms are patentable subject matter, etc.) rather than on *normative analyses* of what is socially desirable and how that might be accomplished (e.g., what aspects of a program are valuable? how might a software developer obtain and preserve competitive advantages? to what extent should the law permit this?). This tendency is understandable and pragmatic. It reflects a natural reluctance to speculate (perhaps pointlessly) on hypothetical changes in the law or to propose changes too readily or too specifically and risk doing harm, rather than good.

The rapid pace of technological change in computer hardware and software complicates the “software debate.”

Evolution of the Software Debate

Throughout the 40-some years of modern programming, computer software has not seemed to fit as easily as computer hardware within the traditional intellectual property framework. Most intellectual property protection for software has come through copyright and trade secret laws, and some through patent law, but software developers and users, the courts, and policymakers have engaged in a continual attempt to sort out *what should or shouldn’ t be*

[14] $ ‘positive analysis’’ refers to an analysis of *what is*. ‘‘Normative analysis ‘‘ is concerned with *what ought to be. In* this context, for example, focus on whether existing law can be interpreted as protecting program command structures would be part of a positive analysis. Focus on whether the program command structure *should* be protected to meet public-interest objectives would be part of a normative analysis.

protected (from a social perspective) and *what is or isn't protected* (according to current law).

By the mid-1970s, this ‘‘software debate’ helped motivate Congress to mandate the National Commission on New Technological Uses of Copyrighted Works (CONTU) to consider the question of how best to treat software. CONTU'S recommendation that copyright protection be explicitly extended to computer programs was reflected in the 1980 amendments to the Copyright Act.[15] But the debate was not put to rest, particularly with regard to the appropriate *scope* of copyright protection. In its 1978 report, CONTU had recognized certain difficulties in applying copyright to software, especially in distinguishing between the copyrightable “expression ‘‘ in a program and the processes or ideas the program implements, which are not copyrightable.[16] CONTU assumed that most copyright infringements in the then-immediate future would be ‘ ‘simply copying, ’ ‘ but recognized that technological advances would raise more difficult questions in determining the scope of copyright.[17]

One such question concerns “reverse engineering’ ‘ of copyrighted programs, especially when it involves translation of object code into higher-level languages. This process is often referred to as “recompilation” (see box 1-A and ch. 4). Discussion of reverse engineering and recompilation brings together a number of copyright issues, including: whether it should be a copyright infringement to read/study a copyrighted digital work in order to extract noncopyrightable subject matter;[18] the extent to which fair use applies to unpublished works; whether the combination of copyright and trade secret laws should be used to achieve protection for noncopyrightable subject matter (ideas, processes, etc.) in copyrighted programs.

As software technologies and markets evolved and grew, so did the controversy concerning appropriate protection for computer programs, computer processes (implemented in software), and algorithms. Since 1981, there have been increasing numbers of patent applications and patents granted for *software-related inventions.*[19] *Over the* past decade, patents have been issued for software-related inventions such as linear-programming algorithms, spell-checking routines, logic-ordering operations for spreadsheet programs, brokerage cash-management systems, and bank college-savings systems. Patent litigation involving software-related inventions and controversies concerning patents for algorithms have become highly visible.[20] These causes of action and invention-specific controversies have focused attention on the appropriateness of patent protection for software-related inventions and algorithms, which present significant problems for patent-system administration. These problems include the incomplete stock of “prior art’ available

[15] Copyright Act of 1976, 17 U.S.C. 101 and 117. See also ch. 2.

[16] *Final Report of the National Commission on New Technological Uses of Copyrighted Works (CONTU)*, July 31, 1978, pp. 18-22. (Referred to by OTA as *CONTU Report.)*

[17] CONTU concluded, however, that these questions should be answered on a case-by-case basis by the Federal courts (ibid., pp. 22-23).

[18] 6‘Ra~g~ ’or ‘‘studying’ a copyrighted work has never, in itself, been a copyright violation. It is only when analysis involves (perhaps requires) the making of a‘ ‘copy’ of the work—usually as an intermediate step in producing a competing work which may or may not be ‘ ‘substantially similar’ and therefore infringing-that the legitimacy of analysis to reverse engineer comes into question.

[19] In this report, OTA sometimes uses phrases like ‘patents for software-related inventions’’ or ‘‘software-related patents” to refer generally to patent protection for inventions implemented in software. (See discussion in ch. 4.)

The U.S. Patent and Trademark Office (pTO) considers terms like “software patents” to be misnomers because they may be interpreted to mean that a computer program per se (i.e., the code) is patentable, as opposed to the underlying computer process. The PTO position is that computer programs per se are not patentable, as opposed to patentable computer processes and algorithms that do not fall into the subject-matter exception for ‘mathematical algorithms.” (M. Keplinger, G. Goldberg, and L. Skillington, PTO, letter to Joan Winston, OTA, Dec. 18, 1989.)

[20] An ‘‘algorithm’’ is a well-defined computational procedure for taking an input and producing an output. Algorithms are tools for solving computational problems-an algorithm describes a specific computational procedure for achieving a desired input/output relationship (see ch. 4).

In the United States, certain types of computer-implemented processes and algorithms can be patented. The U.S. Supreme Court has not ruled on whether computer programs per se are patentable subject matter, but has ruled that computer-implemented algorithms that are deemed “mathematical algorithms’ per se are *not* statutory subject matter. Federal courts have thus held that a computer processor algorithm is statutory subject matter unless it falls within a judicially determined exception like the one for “mathematical algorithms.’

Currently, PTO patent examiners carry out a two-part test for mathematical-algorithm statutory subject matter; the test is intended to be consistent with legislative history and case law. For examination purposes, ‘‘mathematical algorithms” are considered to refer to “methods of calculation mathematical formulas, and mathematical procedures generally,’ and no distinction is made between manmade mathematical algorithms and mathematical algorithms representing discoveries of scientific principles and laws of nature, which have never been statutory subject matter. (U.S. Patent and Trademark Office, “Patentable Subject Matter: Mathematical Algorithms and Computer Programs, ” 1106 O.G. 4, Sept. 5, 1989; also contained in Michael S. Keplinger and Ronald S. Laurie (eds.), *Patent Protection for Computer Software: The New Safeguard* (Englewood Cliffs, NJ: Prentice Hall Law and Business, 1989), pp. 9-42.)

Box l-A—Decompilation

There are three different types of programming languages: *machine language, assembly language, and high-level language. Machine* language programs can be executed directly by the computer, but are relatively difficult to write and understand. Assembly language programs and high-level language programs are easier to write and understand, but cannot be executed directly by the computer. For this reason, programs are usually first written in assembly language or a high-level language, and then translated into machine language so that they can be executed on the computer.

Programs are typically distributed in machine language form. Machine language programs do not have to be assembled or compiled by the user; they are ready to be loaded into the computer and executed. In addition, distribution in machine language form has the side effect that it is difficult for others to look at the program code and understand how the program works. This can help to keep secret those elements which give the program a competitive advantage.

The "recompilation‘‘ issue has arisen because efforts to translate a machine language program back into a more understandable form, such as assembly language or high-level language, may be a copyright infringement because the translation process would involve the making of an unauthorized copy or derivative work. Legal scholars are divided on the question of whether this activity can be excused under the provisions of Section 117 or Section 107 of the copyright law.

Two terms are used to refer to the process of translating a machine language back into a more readable form. *Disassembly* is the process of translating a machine language program into an assembly language program; *recompilation* is the process of translating a machine language program into a high-level program. One issue in the policy debate has been the feasibility of recompilation. There are currently no commercially available decompiler. It appears that the term 'recompilation,' as it is used in the policy debate, encompasses disassembly and any other procedure by which a machine language program is translated into a more understandable form. There area number of disassembler programs available on the market. Translating a machine language program into assembly language is much easier than translating it back into a high-level language.

One view is that limits on recompilation are required in order to encourage the development of original programs. Those who take this position argue that recompilation significantly lowers the cost of implementing "clone" programs. They claim that the original program is decompiled, altered to disguise the copying, and marketed. The clone program can then be sold at a lower price, taking away market share from the original developer, and reducing incentives for the development of new programs.

Others argue that recompilation is a difficult and time-consuming process that does not significantly reduce the cost of developing clone programs. A large disassembled program takes a great deal of effort to understand. In addition, they emphasize that recompilation is required for a variety of other purposes, many of which have a less direct impact on the developer of the program being decompiled. (For more discussion, see ch. 4.)

SOURCE: OTA, 1992.

to patent examiners in evaluating patent applications for processes involving software and algorithms, and the long timelag between patent application and issuance, compared to fast-moving software life cycles (see below and boxes 1-B and l-C). Moreover, some members of the software and legal communities believe that software-related patents will tend to stifle, rather than encourage, technological progress.

Copyright and patent lawsuits have continued to test and explore the boundaries of the current laws.

An incomplete stock of prior art can present significant problems for patent-system administration.

Looking at the scope of current legal interpretations and at possible uncertainties in these laws, some have proposed that modifications to existing structures, or the development of *sui generis* protections[21] are preferable to forcing software to fit

[21] *Sui generis* is a Latin phrase describing a law that is "of its own kind or class.' The Semiconductor Chip Protection Act of 1984 (Public Law 98-620) is a *sui generis* law for chip mask works; it is not part of the patent or copyright laws.

Box l-B—Patent Problem of Prior Art

The quality and availability of the *published (as* opposed to *product)* prior art, or known technology, is often cited as affecting the quality of issued patents (see ch. 2). The U.S. Patent and Trademark Office relies upon its database of prior art to determine whether the invention defined in the patent application meets the patentability criteria set forth in statute. It is against this collection of prior art literature (including earlier patents) that the PTO compares the claimed invention and decides whether the claimed invention possesses the requisite novelty and nonobviousness.

Among the reasons cited for the perceived problem of prior art is the extensive use of trade secret protection for computer software. Unlike patent and copyright protection, trade secret does not require disclosure of information that is the subject of protection. Rather, trade secret requires that the holder of the trade secret make a deliberate effort to maintain the secret quality of the information. Such secret information cannot, by definition, function as part of the "known" technology available as a standard for patentability as required in the patent law. In the course of development of the computer sciences, some advances in the field were published in journals and industry communications, most especially within the academic community. However, many new innovations were not published because they were simply embodied in a product or not considered the type of "invention" which would be the subject of a publication. The prior art that is not the subject of a patent is not always considered to be as rich in the computer science field as in other disciplines. As a result, in large part, the resources available to the PTO for determining obviousness and novelty are more limited than in other fields; in large part the prior art database is limited to software that is already the subject of other patents for software-related inventions. It is therefore often difficult to determine what can be considered the current state of the technology and what can be considered, in the words of the patent statute, ''obvious to the ordinary person skilled in the art' for purposes of determining patentability.

The patent law does not. provide for a free system for third parties to add to the present general stock of prior art. However, it does allow parties to submit art that maybe pertinent to the patentability of particular *issued* patents. As a result, it has been suggested that a private database of prior art be developed by the industry itself, which would allow for free contribution of prior art in an effort to improve the quality of the prior art database available to the PTO and, consequently, of the patents issued.

SOURCE: OTA, 1992.

models more suitable to other types of works and discoveries. However, the majority of legal experts and firms in the industry takes the position that existing structures like copyright and/or patent are adequate to deal with software, that the case law as a whole is evolving appropriately, and that *sui generis* approaches risk obsolescence as the technology changes and lack an established treaty structure providing international protection (e.g., the Berne Convention provides reciprocal copyright protection in over 75 countries). Thus, their tendency is to try to find some way to accommodate specific aspects of software-like protection of user interfaces—within existing structures, particularly copyright. *In OTA'S view, despite the advantages, there are questions as to whether this process of accommodation can-or should---continue indefinitely. With respect to software, there may be a point where it becomes preferable to complement or substitute for the existing structures, rather than extend the scope of copyright to fit certain aspects of software—perhaps, cumulatively, at the expense of other types of works. In* continuing to assess the intellectual property bargain, Congress may conclude that the "balance" for software differs somewhat from that for other copyrighted works.[22]

The stakeholders in the software debate can be categorized in many ways—software creators, soft-

[22] For example, see testimony on the varying concerns of software developers, journalists, and historians regarding fair use of unpublished works at hearings before the House Committee on the Judiciary on H.R. 2372 ("Copyright Amendments Act of 1991"), May 30, 1991 and June 6, 1991.

U.S. law provides that "original works of authorship" are copyrightable subject matter (17 U.S.C. 102(a)). Computer programs are considered to be in the category of "literary works," which are: "works, other than audiovisual works, expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material object, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied, " (17 U.S.C. 101).

Box l-C—Patent Examination Quality and Speed

The prolonged pendency period for patents between time of application and time of issuance has also been of concern in light of the fast-moving nature of the field of the technology.[1] At present, the U.S. Patent and Trademark Office target for pendency is 18 months. The lower range of estimates of the pendency period for software-related inventions is from 18 months to 2 years; some believe that it is more like 32 months from filing to issuance.[2] Stakeholders concerned with software development find this pendency period alarming, given the rapid pace at which the technology advances. They cite the possibility of "landmine patents,' patents which have been pending in the PTO to issue only after others have in the interim unknowingly developed infringing software products. The issuance of such a patent thereafter precludes the making, using, or selling of the software by anyone other than the patent holder.

Also of particular concern is the question of criteria for subject matter patentability under Section 101. Courts and the PTO have struggled with the question of patentability under Section 101 since the late 1960s, and the rapidly advancing nature of the technology forces that debate to continue (see ch. 4). Recently applicants have complained of a proliferation of Section 101 rejections from the PTO, causing some to conjecture that the PTO is implicitly asking the Board of Appeals and the Court of Appeals for the Federal Circuit to issue new rulings on the question of patentable subject matter.

[1] See discussion in ch. 2.

[2] Robert Greene Sterne and Edward J. Kessler, "Worldwide Patent Protection in the 1990s for Computer Related Technology,'' in Morgan Chu and Ronald S. Laurie (eds.), *Patent Protection for Computer Software* (Englewood Cliffs, NJ: Prentice Hall Law and Business, 1991), p. 359.

SOURCE: OTA, 1992.

ware users, large and small commercial software developers, computer hardware manufacturers, educators, students, academic and other software and computer science researchers, to name just a few (see box l-D). Sometimes issues in debate are characterized as conflicts between software producers and consumers, between large and small fins, between major firms and their smaller competitors, between commercial and academic/nonprofit software developers and researchers, or between industry and the general public. Although these characterizations can be helpful in understanding specific issues and positions, one must be cautious about overgeneralizing: for example, some aspects of the controversy over software-related patents are characterized along the lines "large firm versus small developer. ' While it is true that large firms, on average, are more likely to have greater financial and legal resources and more expertise dealing with the patent system, licensing, and litigation, some small firms and entrepreneurs are advocates of patents for software-related inventions and find them extremely advantageous, particularly in attracting investments and in dealing with large competitors.[23] *OTA has found that the most general line of demarcation across stakeholders separates those who perceive significant current financial advantages under the status quo and/or who are relatively confident that their legal and financial resources are adequate to deal successfully with any legal uncertainties or litigation, from those who do not perceive significant financial advantages under the status quo (compared to possible changes or modifications) and/or feel less well-equipped to deal with legal uncertainties or litigation.*

Some well-publicized recent copyright lawsuits have raised issues, which are also being debated outside the courtroom, regarding how far the scope of copyright extends beyond the literal written expression—the program code-to the program's "design,' to the logic underlying a program, and to the program's command structure and interfaces (see ch. 4). *At stake in these decisions is the extent to which copyright (in concert with trade secret law) should be interpreted to give protection to the*

[23] See, e.g., Paul Heckel, "Epilogue: The Wright Brothers and Software Invention, " *The Elements of Friendly Software Design*, 2d ed. (Alameda, CA: SYBEX, Inc., 1991), pp. 223-294; and Elon Gasper et al., "Vital to Small Companies" (letter to the editor), *The New York Times*, June 8, 1989, editorial page.

Box I-D—The Software Debate: Some Stakeholder Groups and Their Concerns

Individual software ***creators and the software industry-Creators of commercial software are*** concerned about their financial viability; an important rationale for intellectual property protection for software is to give commercial software developers adequate market incentives to invest the time and resources needed to produce and disseminate innovative products. The software industry in general is concerned with revenue losses resulting from commercial piracy and counterfeiting and many developers are also concerned about unauthorized end-user copying (see ch. 3). But direct revenue losses are not the only concerns of commercial developers, who also want to gain and maintain a competitive advantage in the marketplace.

One powerful source of market advantage is lead time: the first company out with an innovative computer program benefits from its head start. Trends in software technology, like computer-aided software development, are eroding leadtime advantages. Another potential source of a market advantage is user and/or machine interfaces. In this area however, the industry's goals of expanding the market and a firm's goal of maintaining or increasing market share can beat odds (see chs. 4 and 6).

Software developers, and the industry as a whole, are concerned with access to state-of-the-art knowledge and diffusion of information about programs and programming, so that programmers can build on each others' work, rather than reinvent the wheel (or rewrite a matrix-multiplication subroutine) for each new program. For society as a whole, the pace of innovation maybe speeded up if competitors are able to build on others' advances, rather than allowing an innovator to block others (see chs. 4 and 6). A related concern is reverse engineering of software, particularly for the purpose of understanding the internal construction and functioning of a program (see ch. 4).

Software users—Millions of individuals and thousands of businesses rely on purchased software products for their day-to-day activities and livelihood. As with any product, they care about the price, quality, functionality, ease of use, and variety of software available. Most users, especially business users who rely on software tools for day-to-day operations, are also concerned with the availability of expert support for questions or problems.

The "software workforce" who use and/or create software as part of their jobs want to have transferable skills; thus they are concerned, sometimes only indirectly, with standards for programming languages and external consistency of user interfaces (see ch. 4). (For example, learning a new wordprocessing package is easier if it has commands in common with other packages one already knows.) However, users also want more powerful software

functionality of the program, as opposed to the program code.[24]

Software-related patent suits are also ongoing. This litigation and the recent publicity[25] given to some patents for algorithms[26] have stimulated debate over whether computer processes and algorithms should be patentable at all, or whether they are different enough from other areas of technology that special limitations should apply. Although questions pertaining to patent-system administration are extremely important, the long-term question of whether patent (or patentlike) protection for computer processes and/or algorithms is socially desirable is separate from the related question of how well current U.S. Patent and Trademark Office (PTO) procedures are working now,

In 1990, the Secretary of Commerce established an Advisory Commission on Patent Law Reform. The commission was to examine 13 sets of issues regarding the patent system, including protection of

[24] As Paul Goldstein put it:

> " . . .a computer program is quintessentially a functional work. As a consequence, even the most closely circumscribed definition of a computer program's protectable subject matter will to some degree enable the copyright owner to monopolize the program's function-its 'procedure, process, system, method of operation, concept, principle, or discovery. ' Consequently, the task in defining the scope of a computer program's protectable subject matter is not to distinguish between nonfunctional and functional elements, since function will pervade all elements. Rather, the task in any case is to separate those elements-protectable expression—whose monopolization will not overly inhibit competitors' use of the program's functions, from those elements—unprotectable ideas-whose monopolization will improperly inhibit competitors' use of the programs's function. (Paul Goldstein, *Copyright—Principles, Law and Practice* (Boston, MA: Little, Brown and Co., 1989), sec. 2.15.2, pp. 206-207.)

[25] Sec, e.g., Edmund L. Andrews, ''Equations Patented: Some See a Danger, " *The New York Times,* Feb. 15, 1989, pp. D1,D6; Jack Shandle, "Who Will Weather the Gathering Storm in the Courts?' *Electronics,* August 1989, pp. 67-70; and ' 'Lodging Securities at the Patent Office, ' *The Economist,* Aug. 25, 1990.

[26] See footnote 20, supra.

with improved functions; sometimes the desire for consistent (''standard' interfaces conflicts with ease of use and improved functionality. The software-training and temporary-help industries share these workforce interests; they also are developers of training software.

Thus, software users care about the health of and level of competition in the software industry, as well as having ''common ground' (compatibility) that allows them to use new products with their existing hardware and software (see ch. 4). Users care about having "reasonable" rights (e.g., being able to make a backup copy of an expensive piece of software); some need the ability to modify 'packaged' software in order to use it efficiently or meet other specialized needs. Most businesses and individuals who use software tools to create other products or services want a stable and predictable legal environment so they know what uses are permissible and which are not or must be licensed from developers.

Academic community—Academic and research communities traditionally value free access to and exchange of information (see ch. 5). Academic software/computer science researchers and developers who are motivated by incentives other than commercial potential (e.g., professional prestige, tenure, publication in scholarly journals) tend to view intellectual property protection somewhat differently than commercial developers. Like many small software vendors, many in the academic community are concerned that what they consider to be "overprotection' (e.g., copyright protection for "look and feel' and patenting of software processes and algorithms) might hamper research and long-term growth in their fields (see ch. 4).

In contrast to development of major commercial software packages, "small" software programs to help teach students are developed by faculty in a number of disciplines. The incentives to develop, distribute, and use such ''small' software programs, which are often distributed over academic computer networks, differ significantly from those for commercial software.

As financial pressures mount, universities and their faculties are becoming increasingly interested in commercializing technology and appropriating financial rewards from their intellectual property. At the same time, they are concerned about affordability of the software that students need, both inside and outside the classroom and laboratory.

SOURCE: OTA (adapted from U.S. Congress, Office of Technology Assessment, *Computer Software and Intellectual Property OTA-BP-CIT-61 (New York, NY:* Stockton Press, 1990)).

what PTO terms ''computer-related inventions, ' as well as procedural matters such as a first to file system, automatic publication of applications, and the term of patent protection.[27] The latter questions reflect a concern about the differences between the U.S. patent system and those in foreign countries (see ch. 3). The World Intellectual Property Organization (WIPO) has attempted to harmonize patent laws in member countries. U.S. agreement to WIPO'S draft treaty would entail adjustments in U.S. law including a change from a first to invent to a first to file system (awarding the patent to the applicant who has the earliest filing date) and a change in the term of patent protection from 17 years from grant to 20 years from date of filing.

The judicial system, along with PTO and the Copyright Office, participates in the process of defining the bounds of software protection. The courts help determine requirements for—and scope of—protection under the patent and copyright systems by addressing issues in the course of litigation, whether between parties in the private sector or between the respective government agency and applicants (see chs. 2 and 4). *Given the rapid advancement of the computer sciences, the courts face enormous challenges in resolving the issues raised by the changing technology adequately and in a timely fashion, so as to properly serve the needs of both the industry and society at large.*

Courts must deal with complex and fast-moving technologies.

The problem confronted by the judiciary is twofold. In addressing computer and software issues, courts must deal with technology that is highly complex; the court must find a means to understand

[27] See *Federal Register*, vol. 56, No. 95, May 16, 1991, pp. 22702-22706

extremely technical concepts in order to decide legal issues fairly. In addition, the generally overworked judicial system, with its crowded docket, must render decisions on a technology that often advances faster than such decisions can be reached. These two difficulties raise questions about the courts’ current capabilities to address the issues presented by the software industry.

The Law

The U.S. ‘ ‘intellectual property system’ is a mixture of Federal and State law. Laws concerning copyright, patent, trademark, and the protection of semiconductor chip mask works are under Federal jurisdiction. Laws concerning trade secrets and the misappropriation of confidential business information, and certain limited kinds of ‘‘unfair competition, ’ are under State jurisdiction. Trademarks may be federally registered and/or registered with an individual State; trademark rights may also accrue based on common-law usage. Computer software is distinguished from most other intellectual creations protected by intellectual property law in that it is eligible for protection by patent, copyright, and trade secret laws. Each kind of protection possesses certain strengths and weaknesses, and each protects certain aspects of software in specific ways.

The statutory subject matter of a utility patent is limited to a process, machine, article of manufacture, or composition of matter that is novel, nonobvious, and useful, or to new and useful improvements to these classes of patentable subject matter. In exchange for a sufficiently detailed disclosure of the invention by the inventor, the patent precludes others from making, using, selling or importing components of the patented invention. A patent protects against independent creation, so that to prove infringement the patentee need not show that an invention was ‘ ‘copied’ or acquired through some improper access or means. A U.S. utility patent allows for 17 years of protection for the invention (including application of the underlying idea), during which time the patented invention may be licensed, publicly disclosed, and distributed without altering its legal protection.

A patent protects against independent creation. To prove infringement, the patent holder need not show that an invention was “copied.”

Design patent protection is available for surface ornamentation, configuration, or a combination of both. While the configuration of a useful object may constitute a patentable design, a design dictated by considerations of function is not a proper subject for a design patent. Patent protection for designs is granted for a period of 14 years.

Whether and to what extent software-related inventions are the subject of utility patent protection has been an issue for consideration by the courts and the U.S. Patent and Trademark Office since the early 1960s. The U.S. Supreme Court has examined the issue of patentability of software on a number of occasions, in the cases of *Gottschalk v. Benson, Parker v. Flook,* and *Diamond v. Diehr,* attempting to delineate the limits of patentable subject matter with respect to ‘‘mathematical algorithms. ’

At the same time, the PTO has grappled with several institutional problems, including issues such as: examiner training and turnover, length of pendency periods (from filing to issuance) for patent applications, a backlog of applications, and the quality and extent of the prior art database (see discussion in this chapter, pp. 6-8 and more in-depth discussion in ch. **2).** *In OTA’S view, these problems are serious in that they may affect the quality of the patents issued and create additional burdens for software developers and users (e.g., ‘‘landmine’ patents--see box 1-C).*[28]

U.S. law provides that until the patent is issued, the information contained in the application for a patent remains secret, and therefore may be protected as a trade secret. Information beyond that required for inclusion in the patent to meet the ‘‘enablement’ and ‘‘best mode’ requirements can also be reserved for trade secret protection. Trade secret law protects confidential business information against unauthorized use or disclosure, and is based on statutory and common law and contractual provisions.

[28] As a matter of policy, PTO does not comment on the examination process for issued patents. Because OTA could not be “walked though” application of PTO examination criteria or discuss interpretation of the criteria for specific patents, OTA was unable to make any independent finding on the quality of examination for particular software-related patents.

Like patents, trade secret law can protect the underlying idea of an invention, rather than any particular expression. Trade secret possesses the distinct advantage that, unlike patents, this form of protection does not require any disclosure of information; indeed, trade secret protection is critically dependent on the secret nature of the information, and on the steps taken by the trade secret holder to maintain secrecy. Unlike patent holders, possessors of trade secrets have no protection against independent creations and even subsequent patenting by others of the invention that is the subject of trade secret. (For more on trade secrets, see ch, 2.)

Copyright law, unlike patent and trade secret, protects the expression of an idea rather than the underlying idea itself. Copyright does not extend to any procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied. Rather, copyright is said to protect the expression in the program-which may include such program elements as source code, object code, screen displays, etc. (see chs. 2 and 4).

The evolution of case law in copyright has involved examination of several key issues. Among the most important to software are whether object code as well as source code[29] is protected *(Apple v. Franklin);* whether a program's structure, sequence and organization are protected (e.g., *Whelan v. Jaslow, Plains Cotton Cooperative Association v. Goodpasture Computer Serv., Inc., et al.),* and what such protection implies; and whether the ‘‘look and feel' of the program and its interface is protected *(Lotus v. Paperback Software, Computer Associates v. Altai, Inc.).*

Current law provides for copyright protection for unpublished as well as published works. This is important for computer software, because it facilitates simultaneous use of copyright and trade secret protections. The published version of the copyrighted program can be distributed as ‘‘object code, usually in a machine language that is difficult to read or study. The ‘source code, ' usually written in a higher-level, easier-to-understand computer language, remains unpublished and is often held as a trade secret in order to protect the program's logic and know-how (see below and ch. 4, especially the section on recompilation). However, if the ‘‘ideas' of the program can be ascertained by inspection of the object code, trade secret in such ideas is lost. Also, if and to the extent that recompilation is not a copyright or contractual violation, trade secret protection for the source code can be lost.

Copyright protects the expression of an idea rather than the underlying idea itself.

The Semiconductor Chip Protection Act of 1984 (SCPA) extends legal protection to a new form of subject matter-semiconductor chip mask works—in order to address the problem of chip piracy .[30] The act provides for a 10-year term of protection, and registration under the SCPA is administered by the Copyright Office. Reverse engineering is a defense to a claim of infringement under the act, and provides an exemption from infringement liability in spite of proof of unauthorized copying and striking similarity, so long as the resulting chip product was the result of study and analysis and contained technological improvement. The SCPA provides for remedies similar to those associated with copyright protection, does not allow for criminal penalties, and maintains a higher limit on statutory damages than that provided for in the Copyright Act.

The International Arena

The software industry has become global in character, leading to increasing international efforts to protect intellectual property rights in software. The global nature of the industry and the law is important because of the effect of commercial activities in foreign countries on those in the United States, as well as the similar effect of U.S. activities on those in other countries. In the area of software, as in all industrial and service sectors, companies compete in international and domestic environments. As a result, U.S. legal concepts, definitions,

[29] The computer term ‘‘source code' is often used to refer to a computer program in the language that it was written, usually a high-level language but sometimes assembly language. The term "object code" refers to a program in the form of machine language. See footnotes 57, 58 and 59 (and accompanying discussion) below and also ch. 4.

[30] 17 U.S.C., ch. 9. See also Robert W. Kastenmeier and Michael J. Remington, ‘‘The Semiconductor Chip Protection Act of 1984: A Swamp or Firm Ground?' *Minnesota Law Review,* vol. 70, No. 2, December 1985, pp. 417470.

and policy about software and intellectual property protection for software affect (and are affected by) those of other nations. This report discusses the global software industry and the issue of piracy, and examines various treaties, negotiation efforts and attempts to harmonize domestic and international laws to provide protection for intellectual property.[31]

The Global Software Industry

Although its share of the world software market has declined over the past decade or so, the United States is still the world's leading innovator and producer of computer software.[32] Accurate data on software industry revenues and market shares are difficult to obtain, in part because there are many types of "software industry" data being collected and reported by different organizations.[33] These include data about:

- *software and services,* including processing and professional services, as well as software products;
- *application and systems software,* whether packaged or custom-developed;
- *packaged software,* including applications and systems software;
- *custom software,* professionally developed or extensively tailored to meet a customer's specific needs;
- *personal computer (microcomputer) software,* usually sold as packaged software (although not all packaged software is for microcomputers); and
- *software from ''independent" developers* who are not part of a hardware manufacturer.

This variety of data, collected by different organizations, makes comparison and synthesis extremely difficult.[34] Consistency across types and years is usually not possible when drawing from these published figures.[35]

By ah-nest any measure, though, the United States has a premier role as a producer and a consumer of software. According to one industry estimate, U.S. demand accounted for 52 percent of world *software* consumption in the late 1980s.[36] The U.S. Department of Commerce estimated that global revenues from sales of *software* were more than $65 billion in 1989 and that U.S. software suppliers accounted for more than 60 percent of global software sales .37 According to the Software Publishers Association (SPA), North American revenues from *packaged software for microcomputers* (personal computers) were $4.5 billion in 1990, up 22 percent from 1989.[38]

According to the Computer and Business Equipment Manufacturers Association (CBEMA), in the

[31] Examination of North-South and East-West technology transfer is beyond the scope of this report. For a treatment of global economic competition, with an emphasis on high technology, see U.S. Congress, Office of Technology Assessment *Competing Economies: America, Europe, andthe Pacific Rim, OTA-ITE-498* (Washington DC: U.S. Government Printing Office, October 1991).

Although this study does focus some attention on the relationship between intellectual property and standards (see chs. 4 and 6), a detailed examination of standards is outside the scope of this report. For a thorough treatment of international standards, see U.S. Congress, Office of Technology Assessment, *Global Standards: Building Blocks for the* Future, OTA-ITE-529(Washington, DC: U.S. Government printing Office, April 1992).

[32] Studies in the late 1980s estimated that U.S. producers held a 70 percent share of the global market for software. Estimates by the U.S. Commerce Department now place the U.S. market share at around 60 percent (see ch. 2). Part of the decline in the U.S. shares of software demand and supply has come about naturally as software use becomes more widespread abroad and other nations' software industries develop.

[33] For example, the Software Publishers Association (SPA) collects data on packaged PC software; ADAPSO (The Computer Software and Services Industry Association) reports data on software and services, usually (but not always) from independent mainframe and minicomputer software houses; the Computer and Business Equipment Manufacturers Association (CBEMA) reports data on the information technology industry, including office equipment, telecommunications, electronic data processing equipment, and software and services (including software produced by hardware manufacturers). Moreover, "hardware" companies rdso are software producers-sometimes, like IBM, the largest in the world.

[34] Wherever possible, OTA has specified the type and source of market data and estimates (e.g., "software," "independent software, " "software and services' '); the reader should not expect figures for a given year to "add up" or figures from different sources to be readily comparable.

[35] For instance, a fm whose products include PC application may have at least some of its revenues included in ' 'w-application software, " or ''packaged software; '' it maybe included in 'software and services, ' and may or may not be an 'independent' software house. But a firm whose main products are PC networking software is likely not to be included in data on ''PC-application software."

[36] ADAPSO estimate in Jeff Shear, "Competitive Software Industry Suits Up for Global Hardball," *Insight,* July 10, 1989, p. 38.

[37] Commerce Department estimate cited in *Keeping the U.S. Computer Industry Competitive: Defining the Agenda*, Computer Science and Technology Board (Washington DC: National Academy of Sciences, 1990), pp. 30-31.

[38] Ken Wasch, Nicole Field, and Sara Brown, SPA, personal communication, July 30, 1991.

market for *software and services,*[39] the U.S. industry had domestic revenues of about $93 billion in 1990, a 16 percent increase from 1989 revenues of about $80 billion. Of these domestic revenues, CBEMA estimates that software products accounted for about 45 percent of the total--$42.5 billion in 1990 and $35 billion in 1989.[40] (See ch. 3 for more industry revenue estimates.)

U.S. producers are increasingly challenged by competition from developing software industries abroad, particularly in Europe, where U.S. firms currently hold 70 percent of the PC-software market.[41] With the prospect of a unified market and common standards in Europe in 1993, U.S. firms are facing new competition from Japanese software producers who are establishing themselves in Europe through acquisitions, as well as invigorated competition from European vendors. The United States faces growing competition in Asia from Japanese producers, while software industries in Taiwan and Korea are developing rapidly. And in the United States, U.S. firms face new competition in the domestic market from foreign competitors like the Sony Corp. (see app. A for more on overseas markets and technology initiatives).

Software Piracy

Illegal copying of software results in financial losses to U.S. software firms both directly, through loss of sales and/or royalties, and indirectly, through loss of investment opportunities.[42] Retail piracy—duplication of an entire program for sale by ‘‘pirate’ competitors-and counterfeiting are major concerns of most software companies.[43] *These concerns can be dealt with straighforwardly, at least in theory, by copyright law.*[44] In practice, *enforcement, especially overseas, is difficult. (See* below and ch. 3 for discussion of international treaties and agreements concerning intellectual property and software.)

Although U.S. software developers face increasing competition from foreign competitors, the United States still has a premier role as a software producer.

Estimates of financial losses due to piracy vary. ADAPSO (The Computer Software and Services Industry Association) estimates that one of every two copies of personal-computer software used by corporations in the United States is an illegal copy.[45] In 1990, according to SPA estimates, developers of packaged PC software lost $2.2 billion to piracy within the United States,[46] up from an estimated $1 billion in 1986.[47] Industry estimates of losses from piracy abroad are larger: the Business Software Alliance (BSA) estimates that-looking at *all types of* software-software piracy worldwide causes the

[39] *OTA note:* Revenue reported for ‘software and services“ includes revenues from processing and professional services, as well as from custom and packaged software products.

[40] Oliver Smoot, CBEMA, personal communication, June 30, 1991. See also CBEMA, *The Computer, Business Equipment, Software and Services, and Telecommunications Industry, 1960-2000* (Washington, DC: CBEMA, Industry Marketing Statistics, 1990), p. 100. (Estimates from BDA Assoc. forecast.)

[41] SPA estimate (see ch. 3).

[42] For discussion of revenue losses due to piracy, see U.S. International Trade Commission, “Foreign Protection of Intellectual Property Rights and the Effect on U.S. Industry and Trade, ” February 1988, ch. 4.

[43] *OTA note:* This text uses the phrase ‘‘re~l piracy’’ t. mean unauthorized copying for the purposes of selling the illegal copies or close derivatives; “counterfeiting” to mean passing off illegal copies as the real thing; “end-user piracy” to mean copying by users but not to sell the copies.

[44] Jerome Reichman notes that Anglo-American law tends to use copyright to redress ‘‘pi~cy’ (i.e., slavish imitation) because these countries lack a general-purpose unfair competition law on the European model. Reichman considers that more attention needs to be paid to repression of piracy through international norms of unfair competition law. (Personal communication Sept. 17, 1991.) See Jerome H. Reichman, *Proprietary Rights in Computer-Generated Productions,* paper presented at the WIPO Worldwide Symposium on the Intellectual Property Aspects of Artificial Intelligence, Stanford University, April 1991.

[45] Ronald Palenski, ADAPSO, personal communication, July 10, 1991.

[46] Ken Wasch, Nicole Field, and Sara Brown, SPA, personal communication, July 30, 1991.

SPA’s estimate is based on “average” software prices and an “expected ratio” of software applications to new personal computers purchased in 1990. SPA obtained hardware sales numbers for DOS-based and Apple computers from Dataquest. SPA obtained expected ratios of software to hardware from Apple, Microsoft, and Lotus; these ratios were an expected 3 software applications per DOS machine and 5 per Apple machine. Actual ratios based on software sales were 1.78 for DOS machines and 2.55 for Apple machines. (Nicole Field, SPA, personal communication, Aug. 14, 1991.)

[47] The SPA estimated that microcomputer-software producers lost about $1 billion in sales to ‘ ‘piracy ” (defined by SPA as including both copying for personat use and copying for commercial profit) in 1986. (SPA estimate cited in Anne W. Branscomb, ‘‘Who Owns Creativity? Property Rights in the Information Age,” *Technology Review,* vol. 91, No. 4, May/June 1988, pp. 39-45.)

U.S. industry to lose $10 to $12 billion annually, compared to an estimated $12 billion generated by foreign sales of U.S. software.[48] (For more on piracy and efforts to combat it in the United States and abroad, see ch. 3).

International Treaties and Agreements

The United States is a member of the Berne Convention, the Universal Copyright Convention, and the Paris Convention (patents); the United States is also a party to numerous other multilateral and bilateral agreements. (For a full discussion, see ch. 3.) This section briefly spotlights the provisions of the Berne Convention and U.S. participation in the Uruguay Round of the General Agreement on Tariffs and Trade (GATT) and other international agreements. It also notes the software directive recently adopted by the European Community.

The Berne Convention—-The United States is a signatory to the Berne Convention for the Protection of Literary and Artistic Works. In pursuing its goals of effective, uniform protection of authors' rights in literary and artistic works, Berne employs the principle of national treatment,[49] and limits national treatment through the principles of reciprocity, establishment of minimum rights and automatic protection, and providing for the making of reservations. The Berne Convention protects "literary and artistic works" and does not specifically protect computer programs and databases. However, as the United States protects computer programs as literary works in its copyright law, computer programs are granted protection under Berne in the United States.

When it agreed to Berne, the United States was required to change its copyright law to make it compatible with the treaty through the Berne Convention Implementation Act of 1988. These changes include: abolition of mandatory notice of copyright; maintenance of mandatory deposit requirements; establishment of a two-tier registration system that differentiates between works of U.S. origin and works of foreign origin; a limit on the use of compulsory licenses; a minimum term of protection (life of the author plus 50 years). These changes caused by Berne must be considered or recognized in evaluating options for protecting software in the international arena.

The General Agreement on Tariffs and Trade— Some parties to the Uruguay Round of negotiations of the General Agreement on Tariffs and Trade have attempted to include what they refer to as ''trade-related intellectual property rights' (TRIPs) as a subject of the negotiations. These countries specifically have proposed provisions for protection of intellectual property rights in computer software. The U.S. proposal establishes the Berne Convention as the basis for minimum rights to be granted to authors by contracting parties to the GATT', and then sets forth additional protections provided to computer software and databases in the TRIPs. Provisions in the U.S. proposal reflect the current status of U.S. law protecting computer software.

Other U.S. Participation in International Treaties-In addition to multilateral treaties such as the Berne Convention and the GATT, the United States is party to bilateral treaties with nations in which specific provisions for intellectual property protection for computer software are delineated. For the most part, the United States uses the provisions of the Berne Convention as the bases for these treaties. The United States is also a party to the Universal Copyright Convention, created by the United Nations Educational, Scientific, and Cultural Organization in 1952 to provide an alternative to the Berne Convention that would not require a forfeit of copyright notice requirements.

The United States is party to many multilateral and bilateral intellectual property agreements.

The European Economic Community's Directive on Legal Protection for Computer Software— Following its Green Paper on "Copyright and the Challenge of Technology--Copyright Issues Re-

[4S] Robert W. Holleyman and Lori Forte, BSA, personal communication, July 12, 1991. Estimate includes all types of software, not just PC software. Foreign sales of PC application software are substantially less: SPA estimates that sales of packaged PC application software amounted to $4.5 billion in 1990--up 22 percent from 1989—and that foreign sales amounted to about $2 billion. (Ken Wasch, Nicole Field, and Sara Brown, SPA, personal communication July 30, 1991.)

[49] National treatment requires each member nation to provide the same protection to works of nationals of other member nations as it does to works of its own nationals.

The function, external design, and code of a computer program, as well as the design of its user interface, have been the subjects of intense policy debate.

quiring Immediate Action, ' and after extensive and heated debate, the EC released its Council Directive on the Legal Protection of Computer Programs. In the prologue, the directive asserts the variety and scope of protection given computer software among member states and noted the problems they present to the European common market.

The articles of the directive provide for protection of computer programs as literary works within the meaning of the Berne Convention and establish criteria for authorship and beneficiaries of protection. The directive sets forth specific restricted acts, providing that the author has the exclusive right to reproduce or authorize reproduction of a computer program, to alter, translate or adapt the program, and to distribute the program to the public. The directive provides exceptions to these restrictions, including copying needed to use the program according to its intended use. The directive addresses the issue of recompilation, by allowing reproduction and translation of the code without authorization of the owner under certain conditions and when the information garnered from recompilation is to be used to achieve interoperability. The directive provides a term of protection of life of the author plus 50 years after death. The extent to which the directive addresses the concerns of U.S. manufacturers and reflects US. law responds to the trends in globalization of the industry and the law.

Sofware Technology

In this report, OTA has focused on four elements of a computer programe program function, the external design, the user inteface design, and the program code. Each of these elements has been the subject of an intense policy debate concerning the appropriate level of protection, and the level of intellectual property protection available under current law.[50]

Program Function

Computer programs instruct the computer to perform a series of operations to transform input values to output values. Under current interpretations of patent law, patents may be granted for parts of the program function. The same program may contain many patentable inventions-maybe none at all-depending on whether parts of the program function are novel, nonobvious, and meet the statutory definitional requirements. In applying for a patent, the applicant need not specify each operation performed by the processor, but describes the steps at a higher level of abstraction-e. g., ''storing a set of picture element data in a memory device. ' '[51]

The case law and PTO guidelines indicate that patents may not be granted for a "mathematical algorithm' (see footnote 20, supra). The meaning of the term "mathematical algorithm" has been the subject of considerable discussion,[52] but it appears to refer to a program function that is a "mere calculation." According to PTO guidelines, claims that include calculations expressed in mathematical symbols include a mathematical algorithm.[53] On the other hand, the function is not considered ' 'mathematical' if it can be stated in terms of its operations on things in the ''real world, "[54] e.g., processing architectural symbols[55] or translating languages.[56]

External Design

Another intellectual property question concerns the protection of the external design or "interface" of a program. The external design specifies the inputs and outputs, and the conventions for communicating with a program. For example, a user would

[50] Patent, copyright, and trade secret law all have to be taken into account by software developers.

[51] U.S. Pat. No. 4,197,590.

[52] See Pamela Samuelson, "BensonRevisited," *Emory Law Journal*, vol. *39, No. 4,* fall 1990, pp. 1025-1 154; Donald S. Chisum, "The Patentability of Algorithms, ' *University of Pittsburgh Law Journal, vol. 47, No. 4,*summer 1986, pp. 959-1022; Allen Newell, ''The Models Are Broken, The Models Are Broken!'' *University of Pittsburgh Law Journal,* vol. 47, No. 4, summer 1986, pp. 1023-1035.

[53] U.S. patent and Trademark Office, Computer *Programs and Mathematical Algorithms,* September 1989, p. 8.

[54] *In re Bradley*, 600 F.2d 812 (C.C.P.A. 1979).

[55] *In re Phillips*, 608 F.2d 879 (C. C.P.A. 1979).

[56] *In re Toma*, 575 F.2d 872 (C. C.P.A. 1978).

The user interface specifies conventions for communication between the user and the program.

have to know the specific commands of a user interface, their meaning, and formats for entering data. Other examples of interfaces are communications protocols and operating system calls.

The interface is conceptually distinct from the program code that implements the interface: there are typically many different ways of writing a program to provide the same interface. There has been considerable discussion whether it should be permissible to write a program that has the same external design as a previously copyrighted program. Some believe that intellectual property protection of interfaces is needed, while others believe that it is sufficient that the program code implementing the interface not be copied.

User Interface Design

Courts have been asked to resolve cases that assert protection of communications protocols and operating systems calls, but the type of external design subjected to the most debate has been the user interface. The user interface specifies the conventions for communication between the user and the program. There are a number of different kinds of user interfaces. One is the command language dialogue, in which the user issues commands to the computer through typed commands. If the program is used infrequently, it may be difficult for the user to remember the commands and how they can be used together to perform more complex tasks. "Menu" systems avoid this problem by displaying the command options on a screen; the user can then issue a command by pressing a key indicated as corresponding to a particular menu option, or by moving a cursor on the screen until the appropriate selection is highlighted. Newer interfaces make use of graphics or icons.

Program Code

The program code is protected by copyright: unauthorized duplication of a program except as provided by law (e.g., 17 U.S.C. 107, 117) will nearly always be a copyright infringement. However, an important aspect of the software intellectual-property debate is the degree of similarity that two programs can have, without infringement. The issue is whether two programs should be permitted to have similar ''structure, ' even if not every instruction is identical-i. e., at what level of abstraction above the literal code should two programs be permitted to be the same.

There are three different types of programming languages: machine language, assembly language, and high-level language. Machine language programs can be executed directly by the computer, but are difficult to write and understand.[57] Assembly language programs[58] and high-level language programs[59] are easier to write and understand, but cannot be executed directly by the computer. For this reason, programs are usually first written in assembly language or a high-level language, and then translated into machine language so that they can be executed by the computer.

Programs are typically distributed in machine language form. The program on the diskette is ready

[57] Machine language instructions are patterns of 1's and ()'s which represent digital electronic signals inside the computer. These signals can take on one of two different values; to make it easier to think about what is happening inside the computer, programmers represent one of the values with the symbol 1,' and the other with the symbol "O.' For example, one type of 'addition" instruction for the processor which is used inmost microcomputers may be represented as "OOOOO1OO.' Inside the computer, the pattern of electronic signals corresponding to this pattern of O's and 1's would cause the computer to add two numbers together.

[58] Assembly language makes programming easier by associating a short mnemonic with each type of operation. For example, a programmer using assembly language would represent the: addition instruction discussed above with the word ''ADD. ' Another typical assembly language instruction is "MOV,' which is used to MOVe a piece of data from one place to another inside the computer. Because the computer does not understand the assembly language mnemonics, they have to be translated into machine language instructions using a special program called an assembler. The assembler reads each assembly language instruction and replaces it with the appropriate pattern of 1's and 0's. For example, the ADD instruction might be translated to ''00000100.''

[59] High-level languages (e.g., FORTRAN) are even easier to use and understand than assembly language. The instructions are ' 'English-1 ike, ' as with assembly language, but differ from assembly language instructions in that they are more powerful. Each high-level language instruction does the same job as multiple assembly language instructions. Because the computer does not understand high-level language instructions, they arc translated into machine language instructions using a special program called a compiler. The compiler reads each high-level language instruction and replaces it with the appropriate sequence of machine language instructions. More sophisticated compilers then perform "optimization": they may delete or rearrange machine language instructions in an effort to make the program execute more efficiently.

to be loaded into the computer and executed. Rarely is the program distributed in the high-level language or assembly language in which it was written. Distribution in machine language makes it difficult for others to read the program code to understand how the program works. This helps to maintain secrecy about the elements that give the program competitive value. The ‘‘recompilation” issue is concerned with the legality of efforts to translate a machine language program into a more understandable form such as assembly language or high-level language (see box l-A).

Digital Information and Copyright

‘‘Digital information’ refers to the data stored on computers and in other digital media (e.g., magnetic or optical discs). Computer programs are used to manage and retrieve digital information. Software is necessary for users to access and manipulate digital information stored inside a computer or on storage media. It is difficult, with some modem programming techniques, to distinguish between the computer program and the data the program manages. *Thus, decisions affecting intellectual property and software may also affect digital information and the industries that create and use it. (See* ch. 5.)

Computers are revolutionizing the publishing industry. “Electronic publishing” is now used in the publication of most traditional books, newspapers, and magazines, as well as for delivery of documents to users in digital form. Computer software offers an increasing range of tools for storing, accessing, and manipulating information, Computers make collaboration and multiple authorship easy. Information in digital form is easily copied, transmitted, and modified. *These characteristics make it a good publishing medium, but also raise many intellectual property questions concerning what constitutes a copyrightable work, criteria for evaluating originality and authorship, and new ways of ‘using’ works and compensating authors.*

Due to some uncertainties about the level of protection offered by copyright law to digital data, copyright holders and vendors make use of contracts to attempt to control the uses of digital information by users. Data is often not sold to the user, but is licensed. There is a wide variety of terms and conditions included in these license contracts. Many institutional users of digital information (e.g., libraries and university data centers) complain about the difficulty of managing and complying with the variety of contract terms required by their large collections of data and software packages. There is also controversy about the enforceability of some of these contracts, particularly where vendors have sufficient bargaining power to force terms on the user. (Contracts are also discussed in ch. 2.) Despite provisions of copyright law and license contracts, unauthorized copying of digital information still occurs.

Computer software offers an increasing range of tools for storing, accessing, and manipulating information.

Digital information is not just words and numbers. Anything that can be seen or heard can be digitized, so databases can include music, motion pictures, or photographs of art works. Some databases consist primarily of images. Mixed media or multimedia works are those that package together information in the form of images, sound, and/or text. For example, a multimedia cultural history of the 1960s might include text from newspapers and pamphlets, photographs, recordings of news broadcasts, segments of movies, recordings of music, along with software to access the information, all packaged together in a set of magnetic and optical discs. *There is no specific copyright category for protecting mixed media works.* In* addition, it is not always completely clear what obligations one has to original copyright holders when creating a database of digitized versions of all or part of works that fall under other copyright categories.

If computers have changed the publishing industry, they have also affected libraries. Libraries began embracing computer technology in the 1960s, at frost for administrative tasks like acquisitions and circulation. The first on-line library catalogs began to be developed in the mid-1960s, and many are now available for use by library patrons, offering them much greater flexibility in searching for needed

[60] U.S. law includes eight categories of copyrightable works of authorship: literary works; musical works; dramatic works; pantomimes and choreographic works; pictorial, graphic, and sculptural works; motion pictures and other audiovisual works; sound recordings; and architectural works (17 U.S.C. 102(a)).

works. Some libraries are providing patrons with access to databases of bibliographic citations and full-text journal articles provided by commercial firms, as well as other services like access to electronic mail. These services are most widely available through research libraries at universities, but are increasingly moving into public libraries, where they may be important services for segments of the population that have no other reasonable access to digital information.

Uncertainties have arisen about libraries and digital information. For example, provisions under the copyright law for libraries to lend materials or make preservation copies apply to both printed information and "computer programs' (instructions to the computer) but not necessarily to digital information or mixed media works. Some ways in which libraries might wish to enhance services to patrons, e.g., upgrading on-line catalogs to provide tables of contents and other information from the cataloged books, might be considered to infringe on the underlying works. It is not clear what responsibility libraries may have for patrons' violations of copyright or for contract conditions when patrons have direct access to digital information.

Economic Perspectives

U.S. patent and copyright laws define limited monopoly rights[61] granted to creators of certain classes of ''works and inventions. ' *In this country, these monopoly rights are not viewed as ''natural' or ' 'inherent' rights of creators; rather, they are granted by the government in order to promote the public interest and are designed within a framework involving an economic tradeoff between private incentives and social benefits.* Thus, in the United States, an "intellectual property bargain" underlies the Federal framework for intellectual property law.

The Intellectual Property Bargain in U.S. Law

The rationale for this economic tradeoff-the "bargain" —recognizes that for certain goods, market forces will not necessarily produce the most desirable outcomes from the perspective of society as a whole. These goods will tend to be produced in insufficient quantity or variety because producers are unable to fully realize the gains from investments in creating them.[62] In granting a limited monopoly via copyright or patent, government attempts to compensate for distortions arising from this market imperfection.[63]

The linkage between intellectual property rights and economic benefits to society as a whole has traditionally followed the logic that: 1) intellectual property rights increase innovators' ability to appropriate returns from their intellectual labors; 2) the resulting potential for increased private gains to innovators induces additional innovation; 3) *because of increased innovation,* additional benefits accrue to society as a whole.[64] The U.S. system of patents and copyrights is intended to strike a balance between holders of intellectual property rights and the public at large. This balance involves benefits and costs on both sides: legal protection for intellectual property imposes costs on a society, as well as benefits. These costs and benefits can be monetary (e.g., increased or decreased costs or royalties), or less tangible (e.g., social consequences of stimulated or stifled technological advances). *The specifics of how this balance is maintained--the exact form, scope, and duration of intellectual property rights—may evolve in response to changes in technology, markets, or social values.*

Intellectual Property and Software

Economists have been paying increasing attention to intellectual property and software, but as yet there are no firm conclusions as to what socially optimal protections may be. The lack of precise policy

[61] *OTA note:* In this report, ' 'monopoly" is used in the economic sense and should *not* be taken as synonymous with illegal monopolization of a market or markets. For discussion see F.M. Scherer, *Industrial Market Structure and Economic Performance,* 2d ed. (Chicago, IL: Rand McNally College Publishing Co., 1980), pp. 527-594. As Scherer notes, "Congress [chose] the word 'monopolize' to describe what it condemned, and not some more conventional phrase such as 'obtain or possess monopoly power' '' (p. 527).

[62] Some goods (like information) have the property of nonexclusivity: once the good has been produced and publicly distributed, it is impossible (or prohibitively costly) to exclude any individual from benefiting from it, whether or not he or she pays. Furthermore, consumers' individual self-interests provide strong incentives not to pay for the good, or to undervalue it, in hopes of getting access as "free riders, " See ch. 6.

[63] ''The Congress shall have Power . . . To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries. (U.S. Constitution art. I, sec. 8, cl. 8.)

[64] ''The economic philosophy behind the clause empowering the Congress to grant patents and copyrights is the conviction that encouragement of individual efforts by personal gain is the best way to advance public welfare through the talents of authors and inventors in Science and the useful Arts. " (*Mazer v. Stein,* 347 U.S. 201,219 (1954).)

prescriptions regarding linkages among intellectual property, technological innovation, and social benefits is not unique to software, although economic inquiry is made all the more difficult by the rapid changes in software technologies and markets. Literature on the economics of software is still evolving, along with the broader literature on intellectual property and innovation.[65] Chapter 6 of this report offers a "snapshot' of economic thinking, rather than economists' solution to the problem of how best to balance private incentives and social benefits in a rapidly moving area of technology.

In the United States, an "intellectual property bargain" underlies the Federal framework for intellectual property law.

The economics literature on intellectual property focuses mostly on patent and copyright. *In large part, this focus stems from the nature of patent and copyright." these exclusive rights have been designed within a framework involving an economic tradeoff between private incentives and social benefits .66 The* laws governing trade secrets do not incorporate this kind of explicit tradeoff.[67] Therefore, patent and copyright offer more established economic bases for theoretical and empirical analyses of markets for intellectual property. The bulk of economic analysis on linkages among technological progress, economic welfare, and intellectual property has dealt with the patent system, rather than copyright. Software is remarkable in being a technology for which copyright is so crucial.[68] However, many of the arguments concerning patents and duplication of innovations can be applied to software copyright, especially to issues like copyright protection of interfaces and the appropriate breadth of copyright protection.

As noted previously, this is an evolving literature. Sometimes, the analyses discussed in this report differ in conclusions or policy implications. In particular, some of the economic research done since CONTU suggests policy implications that differ from those of earlier work in the 1950s, 1960s, and early 1970s.[69] These differences result because the economic models incorporate different industry conditions, different types of innovation, and different timeframes. Much of the earlier economic work on intellectual property (mainly patents) focused on cost-saving *process* innovations, while later work looked at *product* innovations. Until after the mid-1970s, most analyses of (socially) optimal patent design focused on *patent term* and assumed static (one-shot) models of innovation. The more recent work focuses on *breadth of protection,* as well as term; dynamic models of innovation include the possibilities of multiple inventors,[70] cumulative innovation, and network externalities. The more recent work, using dynamic models for innovation,

[65] *OTA note:* Unless otherwise specified, OTA uses "innovation" and "innovative activity'' in this chapter to refer to R&D and other creative processes producing scientific and technological advances, whether the form of these advances would legally be considered copyrightable, patentable, or neither. In reviewing the economics literature on this topic in ch. 6, OTA uses the authors' terminology.

[66] Private incentives are expected to arise from the right holder's limited monopoly powers; social benefits are expected to include additional benefits to society from the induced disclosure and/or dissemination of innovations and technological advances.

For discussions of this balancing between private incentives and social benefits in the "intellectual property bargain," see *intellectual Property Rights in an Age of Electronics and Information,* op. cit., footnote *2. See also* Paul Goldstein, op. cit., footnote 24, sees. 1.1 and 1.2.

[67] See Stanley M. Besen and Leo J. Raskind, "AnIntroduction to the Law and Economics of Intellectual Property, ' *Journal of Economic Perspectives, vol. 5, No.* 1, winter 1991, pp. 3-27, esp. p. 23,

The rightful possessor of a trade secret does not have an exclusive right to use the secret information, and the law only provides for legal remedies when the secret is lost through breach of contract or ''impmper" means of discovery (e.g., industrial espionage). A trade secret may be maintained indefinitely. See ch. 6 and also David Friedman et rd., "Some Economics of Trade Secret Law," *Journal of Economic Perspectives,* vol. 5, *No.* 1, winter 1991, pp. 61-72.

6s However, there is also a well-developed literature dealing with economic welfare, copyright, and consumer copying of journal articles, music, software, etc. (see the final section of ch. 6 on home copying).

69 As described by Sidney Winter, the pendulum of opinion on the ''optimal'' term of protection (e.g., whether increasing or decreasing the term Of patent protection would be more socially desirable) has swung back and forth over the years. For his discussion of changes in economic thinking about the term and strength of protection, see Sidney G. Winter, "Patents in Complex Contexts: Incentives and Effectiveness,' in Vivian Weil and John W. Snapper (eds.), *Owning Scientific and Technical information (New* Brunswick NJ: Rutgers University Press, 1989), pp. 41-43.

For another discussion of economics literature on innovation, see Robert P. Merges, "Commercial Success and Patent Standards: Economic Perspectives on Innovation, " *California Law Review, vol. 76,* pp. 803-876, 1988.

[70] In multiple-inventor "patent races, " the rate of R&D spending affects the probability of invention.

suggests shorter, rather than longer, terms of protection.[71]

Software Industry and Technology Changes

Computing technologies and the software market have also evolved since the mid-1970s (see chs. 3 and 4). Because of timing, CONTU and its analyses could not foresee the time when powerful computers could be in every office and every home and individuals would be able to create and use sophisticated and valuable software outside large organizations, or the full impact of these changes a decade later.

CONTU saw the software market shares of hardware producers being "steadily eroded" by independent software developers and expected this trend to continue.[72] But, despite vigorous growth by independents, computer-hardware firms retain a major share of the software market. (In terms of revenues, IBM is the leading U.S. software producer overall and the largest producer of packaged software in the world—see ch. 3). In CONTU'S analysis,[73] the software industry was characterized by easy, rapid entry by small fins-a viable ''cottage' industry of small developer-vendors.[74] Although there are still thousands of small companies, the industry today is moving away from this picture. Significant changes, which may affect the way in which Congress sets the balance for software, include:[75]

- the ''PC Revolution' and explosive growth in markets for personal computers and packaged software;
- widespread use of computers and software by nonprogrammers and the corresponding market importance of user interfaces;
- increased barriers to entry by small firms and a trend toward centralized software-publishing houses that acquire rights to software and then distribute and market it, paying royalties to the program developers;
- maturity of the software industry and increasing firm size (through growth, acquisition, and consolidation); and
- increasing industry concentration, especially when considering submarkets like PC applications.

Issues and Options

In this report, OTA focuses on the various ways in which current U.S. copyright, patent, and trade secret laws apply to four key elements of computer program development—the *program function, the external design, the user interface design,* and the *program code. Our* study has examined the technology behind each of these aspects of the development process, the application of current intellectual property laws to each of the four elements, and the policy issues and arguments associated with them, in terms of current law and in terms of possible modifications to the existing intellectual property system.

Technological Challenges for Copyright Law

OTA finds that treating computer programs as literary works under copyright offers straightforward remedies for the literal copying of program code, although enforcement remains a problem, **especially overseas. (See** ch. **3.) OTA** also finds, however, that the functional aspects of computer programs pose difficult questions for application of the copyright law.[76] One important question is the extent to which copyright (particularly, in concert with trade secret law) should protect the functionality and design of a program. OTA finds that the traditionally "fuzzy" line between idea and expression in copyright law is confounded by the need to determine an appropriate scope of protection in order not to provide coverage for the program procedure, process, system, method of operation, or concept, con-

[71] See Winter, op. cit., footnote 69

[72] *CONTU Report*, op. cit., footnote 16, p. 24. (Quoting from the economic analysis prepared for CONTU by the Public Interest Economics Center, *An Analysis of Computer and Photocopying Copyright Issues From the Point of View of the General Public and the Ultimate Consumer* (Washington, DC: June 1977), p. IV-13.)

[73] Ibid., p. 23.

[74] Public Interest Economics Center, op. cit., footnote 72, p. IV-5.

[75] For discussion, see box 6-A in ch. 6.

[76] See also *Intellectual Property Rights in an Age of Electronics and Information,* op. cit., footnote 2; and Paul Goldstein, op. cit., footnote *24,* sec. 2.15.2, pp. 206-207. But see Morton David Goldberg and John F. Burleigh, "Copyright Protection for Computer Programs: Is The Sky Falling?' *AIPLA Quarterly Journal, vol. 17, No. 3, 1989;* and Antbony L. Clapes et al., "Silicon Epics and Binary Bards: Determining the Proper Scope of Copyright Protection for Computer Programs, ' *UCLA* Law *Review,* vol. 34, June-August 1987.

trary to the intent of the current law (17 U.S.C. 102(b)) .77 This key software-copyright question is manifested (among other places) in the debate over application of copyright to features of a program’s external design and/or user interface, and in the debate over reverse engineering and decompilation (see above and ch. 4).

Technological and Institutional Challenges for the Patent System and PTO

Patent protection is used extensively to protect software-related inventions in the United States, Japan, and Europe. Many of the major commercial developers of software (including firms like IBM that tend to be thought ofas‘‘hardware’ companies) are extensively using patent protection for software-related inventions, As a result, patent protection is of importance to the U.S. software industry, both domestically and in the global market.

The protection of software-related inventions and algorithms by patent[78] is a fairly recent development and is controversial.[79] (See chs. 2 and 4 and boxes 1-B and l-C.) The case law and PTO guidelines indicate that patents may not be granted for a specific kind of program function called a ‘ ‘mathematical algorithm’ (see footnote 20, supra and discussion in ch. 4). The meaning of the term ‘‘mathematical algorithm’ has been the subject of considerable discussion and debate. Moreover, some academics and members of the software community perceive that the technology[80] and economics and industry structure[81] of software development make patents inappropriate for software-related inventions.[82] Many other members of the same communities, however, disagree with this perception and consider that patents for software-related inventions are appropriate for the industry and are in the public interest.[83] Furthermore, the latter argue that the ‘‘disadvantages’ perceived by critics are fully addressable by changes in the operation of the PT0.[84]

OTA finds that the PTO faces considerable challenges in examining applications for software-related inventions. At the same time, there ap-

[77] ‘‘Somc concern has been expressed lest copyright in computer programs should extend protection to the methodology or processes adopted by the programmer, rather than merely to the ‘writing’ expressing his ideas. Section 102(b). . .is intended, among other things, to make clear that the expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law. ’ (U. S.C.A. 17 sec. 102, “Nature of Copyright, ” p. 17, from House Report No. 94-1476, Committee on the Judiciary.) See also footnote 24, supra.

Congress has already (17 U.S.C. 117) limited copyright holders’ exclusive rights for computer programs.

[78] See footnotes 19 and 20, supra.

[79] ‘‘Algorithms (also known technically as the partially recursive functions) form the essence of software. Increasingly we are seeing lawsuits or threats of lawsuits claiming patent infringement for the mere use of software on a typical computer, e.g., for such functionality as public key encryption compression and cursor blinking. If patent law establishes that such suits are justified, that will mean to me that algorithms do have patent protection. To underscore that algorithms do have the same unpatentability as scientific principles, I think patent law should be clarified to the effect that a patent is never infringed merely by the use of software on a computer. ’ (Robert S. Boyer, Professor of Computer Sciences, University of Texas at Austin, letter to Harry F. Manbeck, Jr., Assistant Secretary and Commissioner of Patents and Trademarks, Sept. 22, 1991.)

so See, e.g., Daniel J. Bernstein, University of California, letter to E.R. Kazenske, PTO, July 10, 1991 (Response to Request for Comments for the Advisory Commission on Patent Law Reform). Using issued patents from the field of data compression as examples, Bernstein discusses his views that PTO examiners have failed to detect “mathematical algorithms” in claims, that PTO is not correctly evaluating criteria for nonobviousness and equivalence of patent claims for algorithms, and that “software patents” are beginning to damage the software industry. See also ch. 4.

[81] For example, entry barriers are said to be lower for software than for commercial biotechnology, which has led to an industry with more small entrepreneurs for whom the administration of patents would be difficult, especially given that one program may contain (or infringe) many patented processes. See ch. 6.

[82] For arguments against ‘‘software patents, ’ see Pamela Samuelson, op. cit., footnote 52; and Richard Stallman and Simson Garfinkel (The League for Programming Freedom), ‘‘Against Software Patents, “ Communications *of the ACM, vol. 35, No.* 1, January 1992, pp. 17-22,121.

Among the problems noted are an incomplete prior art and insufficient examiner training, leading to difficulties in examination and issuance of patents that are not novel and/or nonobvious. Another problem noted is that the pendency period is long compared to software-development cycles, so that ‘‘landmine’’ patents can issue. (See chs. 2 and 4.)

[83] See, e.g., Paul Heckel, op. cit., footnote 23; John L. Pickett, President, CBEMA, letter to E.R. Kazenske, PTO, in response to Request for Comments for the Advisory Commission on Patent Law Reform, July 15, 1991; Esther Schachter, Chair, ADAPSO Intellectual Property Committee, letter to E.R. Kazenske, PTO, in response to Request for Comments for the Advisory Commission on Patent Law Reform, Sept. 4, 1991; and Robert G. Sterne, letter to Paula Bruening, OTA, Oct. 6, 1991. Steme argues that many inventive aspects of software-related inventions that are protectable by patent cannot be protected adequately and/or effectively and/or at all by copyright and trade secret (ibid., p. 2).

S4 John L. Pickett, op. Cit., footnote 83. CBEMA rogues that perceived disadvantages such as ‘ ‘bad patents’ anticipated by the prior art, examiners’ difficulties in deciding questions of novelty and nonobviousness, and vulnerability to patents issuing after long delays *IOTA note:* i.e, ‘‘landmine’’ patents] are fully addressable by changes in operation of the PTO. (Ibid., p, 2.)

pears to be some variance-or, at least, uncertainty on the part of observers outside PTO—in how PTO examination guidelines for subject-matter determinations are being applied. A long series of (often inconsistent) court decisions has led to a situation in which some types of software-related inventions are patentable while others are not. Applying the PTO guidelines for distinguishing patentable and nonpatentable types of inventions is a complex part of the examination process and one that outsiders find difficult to understand and/or predict.[85] (As a matter of policy, PTO does not comment on the examination process for issued patents. Because OTA could not be "walked through' application of PTO examination criteria or discuss their interpretation for specific patents, OTA was unable to make any independent finding on the quality of examination for particular software-related patents.) In addition, the PTO has an incomplete database of "prior art" for software-related inventions.[86] This makes it even more difficult for examiners to judge whether an application describes a "novel" and "nonobvious" invention. Filling in the gaps in the database of prior art maybe difficult., because so much of what would constitute the ‘‘prior art’ has historically been in the form of products, not literature or issued patents.

To address the issue of examination quality, the U.S. Patent and Trademark Office is currently reviewing the nature of the qualifications required for examiners and has recently completed the first phase of its reclassification of the software arts. This reclassification process involves the creation of a new Class 395, "Information Processing System Organization.’ ’[87] To make searching easier, this new class will have a larger number of subclasses than the old Class 364, each encompassing a particular area of technology-e. g., database and file management systems or artificial intelligence. PTO intends that examiners will specialize in one of these subclasses. In addition, the Secretary of Commerce established an Advisory Commission on Patent Law Reform** to examine administrative and procedural challenges facing the patent system.

OTA finds that "filling in" the prior art database (patent and nonpatent) is extremely important, as one means of improving the quality of examination. OTA also finds that improving electronic search and retrieval capabilities for the PTO’S own database is critical, because it is used by the PTO’S own examiners during the application process and by the public. In September 1991, PTO reported that it is unable to provide statistics on the number of patents issued for software-related inventions (e.g., patents for computer processes and algorithms), which PTO refers to as ‘‘computer-implemented process patents. ’[89] Despite the intense controversy and policy focus on these areas of art since *Diamond v. Diehr, PTO* reported to OTA that it has no provisions for flagging, cross-referencing, or otherwise efficiently monitoring and reporting prosecution, issuance, and litigation for these types of patents, except through time-consuming manual search, review, and selec-

[85] ‘‘~~so finds the guidelines themselves to be a reasonable reading of case law but questions how the PTO is actually administering the stated Guidelines. Enough ADAPSO member companies have noted a substantial increase in subject-matter rejections on computer program-related claims as to constitute a new trend. . while these rejections cite Section 101, this new practice has no obvious basis in the statute itself and does not represent the kind of result that we believe the Guidelines would lead one to expect. " (Schacter, ADAPSO, op. cit., footnote 83, pp. 15-16.) Others have questioned how examiners interpreted the PTO guidelines in *issuing certain* patents. (See, e.g., Brian Kahin, ‘‘The Impact of Software Patents, ’ *EDUCOM Review*, winter 1989, pp. 26-31 .)

[86] See Jeffrey M. Samuels, ~@ Commissioner of patents and Trademarks, testimony at Hearings on Computers and Intellectual Property, Mar. *7*, 1990, U.S. House of Representatives, 101st Congress, 1st and 2d Sessions, Subcommittee on Courts, Intellectual Property, and the Administration of Justice, Committee on the Judiciary, Serial No. 119, pp. 325-354. According to Samuels' testimony, ‘‘We are concerned that some computer processes that are sold or are in use are not fully described in the published literature or readily evident from use of the process. The sale or use of the process is evidence that the process is not new and should not be protected by a later filed application. Regrettably, there is not an efficiently searchable record of this type of prior art, not only for computer-related inventions but for all inventions. ’ (Ibid., pp. 337-338.)

[87] Gerald Goldberg, Director, Group 230, PTO, personal communication, Oct. 18, 1991.

[88] For further discussion of the Advisory Commission on Patent Law Reform, see pp. 10-1 1 of this chapter, and ch. 2.

[89] See Jeffrey M. Samuels, Acting Commissioner of Patents and Trademarks, letter to Congressman Robert W. Kastenmeier, Chairman, Subcommittee on Courts, Intellectual Property and the Admini‘stration of Justice, Nov. 1, 1989, answer to question 1: ‘‘Computer processes are not classified within USPTO’s patent classification system in any readily identifiable set of classes and subclasses. ’ See also Harry F. Manbeck, Jr., Assistant Secretary and Commissioner of Patents and Trademarks, letter to Joan D. Winston, OTA, Sept. 11, 1991, p. 1 (". . .it is not possible to generate reliable data in response to questions directed to computer-implemented process patents through the manipulation of existing PTO databases’ ‘).

tion from various large patent subclasses.[90] OTA found, however, that the private sector has had more success in developing such statistics and classification schemes. Electronic Data Systems Corporation (EDS) has compiled statistics on patents issued for software-related inventions during the years 1972 through 1989 by examining notices published in the Official Gazette of the Patent and Trademark Office. EDS obtained similar statistics for 1990 by reading official database tapes purchased from the PTO. With the 1990 data, an artificial intelligence technology (involving a natural language interface created by EDS) was used to characterize each patent and determine which involved software-related inventions. Using this method, EDS found that 576 patents were issued for software-related inventions during 1990. (According to EDS, this technology will be used to reexamine the statistics initially gathered for 1972 through 1989. For more on the EDS statistics, see table 2-1 and accompanying discussion in ch. 2.)

These shortcomings in the PTO database affect searches conducted by or for the public. Therefore, while "filling the gaps" in the prior art will be useful, these steps should be taken in conjunction with measures to improve electronic search and retrieval and provide statistical information for use within PTO and for reports to Congress. PTO is still in the process of deploying its automated patent system and has also begun to reclassify patents in the computer arts (see discussion inch. 2, pp. 54-56). As part of its oversight, Congress may find it useful to receive statistical profiles of patent activity in this and other important areas of technology.[91] Therefore, Congress may wish to determine what improvements in statistical reporting by field of technology will be part of this automation and reclassification.

International Dimensions

Computer software markets are international, as are software research and development. Although software markets and industries abroad are growing, the United States currently remains a major force in this international market (see ch. 3). The issues and questions facing Congress are more complicated because of the "global" nature of software. The balance struck in the intellectual property bargain' cannot ignore increasing foreign competition in overseas markets-and within U.S. borders.[92]

The paradigm of software as or akin to a literary work under copyright is the keystone of existing international copyright agreements. However, foreign countries may treat software differently in some respects, even with a copyright framework. For example, the European Community's software directive includes specific provisions concerning certain aspects of reverse engineering, referred to in the directive as "recompilation" (see ch. 3).

The level of patent protection offered by foreign countries for computer software varies: while some may not protect software per se, they may grant protection for processes that include software, or if a program is claimed in conjunction with a method or computer. To the extent that the market for software is global, policy decisions about patent protection for software reflect these differences in

[90] From letter t. Joan D. Winston, OTA, from Harry F. Manbeck, Jr., Assistant Secretary and Commissioner of Patents and Trademarks, Sept.11, 199 1: ''In sum, PTO is not able, through its existing databases, to respond to OTA's request for data concerning 'computer process and algorithm' patents. To compile data with which to respond to OTA's range of questions involving computer-implemented process patents would require a manual search of many technology classes, a thorough review of the claimed invention, and the investment of hundreds of staff weeks. '

OTA had requested statistics from PTO concerning prosecution, issuance, and litigation of patents for software-related inventions. In its request, OTA had asked for statistics on what it referred to as "computer process and algorithm patents. " OTA staff asked to meet with PTO staff to discuss the request-particularly, to explore PTO'S suggestions for alternative formulations of OTA's questions in order to facilitate a meaningful response. (Personal communications with PTO staff March-June 1991 and letter from Joan D. Winston, OTA, to Lee Skillington, Office of Legislative and International Affairs, PTO, June 24, 1991,)

No such discussion or suggestions were provided and PTO responded that, ''As a general matter, the denomination 'computer process and algorithm' patent bears no direct correlation to PTO policy or practices. . patents issued for inventions involving computer-implemented processes are not classified in a single technology class or subclass. . and have not been otherwise 'flagged' to enable the retrieval of the requested data. . even if we were to limit the task to Subclass 364/200 and Subclass 364/900, the task would still be a significant one. , .[that] would probably exceed 30 staff weeks. " (Manbeck, op. cit., footnote 89, enclosure item 3.)

[91] For example, OTA had asked PTO for data on patents and patent applications for computer processes and algorithms: the numbers Of such patents issued from 1974-present, cumulative numbers of patents in effect during this period, average tendencies, examiner rejections appealed to PTO and the courts, etc. PTO reported that it was unable to provide this information because it would have required ''hundreds of staff weeks' to prepare. (Manbeck, op. cit., footnote 89.) In 1989, the House Subcommittee on Courts, Intellectual Property, and the Administration of Justice asked for similar information as part of its oversight of computers and intellectual property; PTO reported that it could not provide it. (See footnote 89, supra.)

[92] For example, the United States is a signatory to the Beme Convention and Universal Copyright Convention. Under these agreements, our domestic copyright law applies reciprocally to foreign copyright holders who are nationals of convention members (see ch. 3 and footnote 49, supra).

legal systems. The World Intellectual Property Organization's Committee of Experts on the Harmonization of Certain Provisions in Law for the Protection of Inventions is considering a draft treaty that would provide for modifications of general aspects of patent system, not specific to software, including a first-to-file (an application with the patent office) system, and establishment of a term of patent protection of 20 years from the date of filing a patent application.

Digital Information and Copyright

OTA finds that many of the issues of concern with copyright and computer software also apply to digital information. For example, copyright provides remedies for the literal copying of digital information, but as with software, enforcement may be a problem. There appear to be few technological or other remedies to prevent unauthorized copying, except within closed systems.

It would be helpful for both publishers and users of digital information if some aspects of fair use of digital information under the copyright law were clarified. Because of uncertainties about users' rights to "download" or make copies of information, providers of digital information rely on contracts to limit customers' uses of information, and do not sell information to customers, but merely authorize certain uses. On-line digital information may pass through several intermediaries between the publisher and the end user--distributor, database service, library-making contracts less effective for controlling end-user practices.[93]

Some aspects of fair use also remain unclear with regard to libraries. For example, while guidelines have been developed for libraries' making archival copies of books or of computer programs, no mention is made of rights to make such copies of databases or other information in digital form. There is also some question as to how far libraries can go in enhancing the content of their on-line information retrieval tools by including more information from the original works (e.g., the table of contents or the index). While such enhancements are now feasible and seem a logical step, some hold that such enhancements may infringe on the copyrights of the underlying works.[94]

Digital information includes multimedia or mixed media databases, which may include images, music, text or other types of works. The status of mixed media works under copyright is not clear. Mixed media is a fairly new concept; the acquisition of rights to convert copyrighted works to digital form for incorporation in mixed media databases is often difficult because conventions and standards for royalties do not yet exist, nor are there organizations of rights holders to collect the royalties.

Software Technology, Industry Structure, and the Future

Software and computer technologies are fast-moving and complex. The software industry and the discipline itself are maturing: the software industry structure has changed since CONTU and the costs of successful market entry are rising. Although there continue to be many viable small firms and entrepreneurs, the industry is moving away from the model of a cottage industry (see box 6-A inch. 6). With the vast increase in numbers and types of software users, user interfaces have become increasingly important, both to users (in terms of ease of learning and use, performance, productivity) and to developers as well (in terms of value in the marketplace and market share).

Despite the advantages of incremental accommodation within the current structures, especially in terms of established case law and reciprocal international protection, OTA finds that there may be a point where it is in the public interest to develop new law(s) either ***to complement*** the existing framework or ***to substitute for*** copyright and/or patent protections for software, rather than continue incremental accommodation. Congress may eventually find that the best means for achieving policy objectives with respect to software are different from those used for other

[93] Brian Kahin, "Contract and Fair Use Issues in Downloading: Subcommittee Report, Committee 702 (Databases), ' in *Section on Patent, Trademark and Copyright: Committee Reports* (Chicago, IL: American Bar Association 1989), pp. 405411.

[94] Mary Jensen, Director, University of South Dakota Law School Library, personal communication, Feb. 8, 1991.

types of works, based on the intellectual property bargain.[95] In its deliberations, Congress could draw upon public input from many economic and social sectors[96] in assessing the net impact of new rights that might be created as alternatives to a strategy of accommodation within the current structure.

Some commentators favor *sui generis* approaches, either to complement or substitute for current software protections. Most proponents of a ' 'substitute' *sui generis* law seem to envision a modified copyright approach.[97] That is, a copyrightlike registration would continue, but the term of protection and the bundle of rights would be modified to conform to what is considered to be the needs of software. In the Semiconductor Chip Protection Act (SCPA), a modified copyright approach is used to protect chip mask works from copying.[98] Some early proposals for the protection of semiconductor chips had recommended amending the Copyright Act, but a *sui generis* approach was chosen to avoid distorting traditional copyright principles for other categories of works .[99]

Software features and advances that may be valuable and beneficial to society may not be traditionally patentable or copyrightable subject matter. Therefore, OTA finds that Congress may wish to consider periodically whether there is some public advantage in giving limited rights for incremental software advances that would not be patentable or for aspects of program functionality that fall outside copyrightable subject matter. In this case, the subject matter, scope, term, and exemptions from infringement of a "complementary' *sui generis* law could be carefully tailored to fit the characteristics of the technology and its uses.[100]

One intellectual property scholar has suggested that software is an example of a "legal hybrid" that falls between patent and copyright.[101] To encourage innovation, these hybrids are thought to require some kind of protection to ensure lead time. However, for these hybrids, it is thought that patents will usually protect only a small portion of the innovation, and the ''powerful reproduction rights and long term of protection [of copyright] implement cultural policies that are largely irrelevant to the needs of a competitive market. [102] According to this logic, in addition to fading to protect innovation

[95] For example, Congress might wish to consider whether the trend toward more use of patent protection, absent alternatives to protect program functionality, affects the public-interest "balance" in terms of equity for small/large software firms and for those with many/few legal and financial resources: What will be the effect on end users and the public at large? Will ''stronger' protection for software (e.g., patents precluding commercial exploitation of independent program inventions, copyright protection for the design of user interfaces) spur innovation stifle it, or have no real effect overall? Will it disproportionately disadvantage individuals and small firms versus large (or rich) corporations?

[96] See, e.g., the discussions of public input in crafting the SCPA in Robert W. Kastenmeier and Michael J. Remington, op. cit., footnote 30, esp. pp. 424-432 and 442-459; and Richard H. Stem, "Determining Liability for Infringement of Mask Work Rights Under the Semiconductor Chip Protection Act, " *Minnesota Law Review,* vol. *70, No. 5,* December 1985, pp. 271 et seq.

[97] See Pamela Samuelson, op. cit., footnote 52, esp. pp. 1148-1153; and Richard H. Stern, "The Bundle of Rights Suited to New Technology,'' *University of Pittsburgh Law Review,* vol. 47, No. 4, p. 1229.

A modified copyright approach has previously been used for the protection of semiconductor chips (see below and ch. 2). There have been a number of proposals to protect industrial designs using a modified copyright approach, but these have not been enacted into law. For a discussion of industrial design protection, see ch. 2.

[98] The SCPA Uses a modified copyright approach to protect the topography of integrated circuits against copying. There is 110 patentlike examination process; the ''mask work' is registered with the Copyright Office. However, the SCPA has a novelty standard somewhat higher than the mere ''originality' standard of copyright law: protection is not available for a mask work that ''consists of designs that arc staple, commonplace, or familiar in the semiconductor industry or variations of such designs, combined in a way that, considered as a whole, is not original'' (17 U.S.C. 902(b)(2)). The bundle of rights is also somewhat different from that granted under copyright law, and copies of the ''mask work' made in the course of reverse engineering are not infringing (17 U.S.C. 906(a)). Finally, semiconductor chip protection differs from copyright in that the turn of protection is only 10 years.

[99] See Kastenmeier and Remington, op. cit., footnote 30, pp. 424-430 and 442-444 and H.R. Report No. 781, 98th Congress, 2d Sess., 1984, pp. 5-11.

[100] For example, use of copyright mandates a long term of protection for software, no provision for compulsory licensing, and limited exemptions from infringement (e.g., secs. 107 and 117). Patent has a shorter term, but many program features may not be patentable subject matter; although reverse engineering is allowed, independent invention is not a defense to claims of patent infringement.

The SCPA, which is not part of the copyright law but is "in harmony" with it, was tailored with a shorter term, technology-specific subject matter (original mask works), and explicit exemptions for reverse engineering that differ from copyright's fair-use exemptions. See Kastenmeier and Remington. op. cit., footnote 30, pp. 445-452,

[101] J.H. Reichman, "Computer Programs as Applied Scientific Know-How: Implications of Copyright Protection for Commercialized University Research, " *Vanderbilt Law Review, vol. 42, No. 3,* April 1989, p. 655.

[102] Jerome H. Reichman, ''Proprietary Rights in the New Landscape of Intellectual Property Law: An Anglo-American Perspective, ' study prepared for the International Literary and Artistic Association, Congress of the Aegean Sea II, June 19-26, 1991, p. 54.

properly, attempts to use existing laws for these hybrids risk distorting the existing laws.[103]

The CONTU report had noted that patent protection for software was limited, and that some additional form of protection would be required. However, the Commission did not recommend a *sui generis* approach, concluding that copyright was appropriate.[104] The argument that computer programs are ‘‘useful articles’ or otherwise fall outside the range of statutory subject matter was rejected by CONTU[105] and has been consistently rejected by the courts.[106] Proponents of the continued use of copyright law contend that copyright, as interpreted by the courts, is working well,[107] that a new system would create unacceptable uncertainty, and that existing international agreements provide a framework for the protection of computer programs in other countries (see ch. 2).

Policy Choices and Options

OTA has identified three principal policy areas that Congress may wish to address. These are:

1. difficulties that the functional aspects of computer programs present in determining the appropriate scope of copyright protection for programs;
2. difficulties in determining the scope of patent protection for software-related inventions and algorithms and the challenges facing the U.S. Patent and Trademark Office in these areas of art; and
3. complications facing libraries and developers and users of digital information, especially mixed media works encompassing several different categories of’ copyrightable works.

As the preceding sections and body of this report detail, these principal areas encompass a variety of issues. For each, Congress haas fumdamental choices.

The first of these choices is to act or not to act. Not acting continues the status quo in terms of statute, and allows the continued evolution of the case law, but does not assuage uncertainty. On the other hand, taking action may reduce some uncertainties but add others, especially if additional bodies of case law and new international agreements had to be developed (e.g., for a *sui generis law).* This choice is not a static decision—Congress may wish to periodically reevaluate the choice to take action or continue the status quo. In so doing, Congress can draw upon input from the broad communities of stakeholders in these issues. These stakeholders include the computer and software industries; members of the academic, research, and library communities; private, corporate, and institutional software users; the software-using workforce; and the public at large (see above discussion on the evolution of the software debate and box l-D).

The second choice, if Congress determines that action will be in the public interest, is the timeframe for action. If sufficient information concerning an issue and how to resolve it unambiguously is available, Congress could act in the near term. In instances when institutional problems and alternative courses of action are well-defined, near-term actions may be both appropriate and necessary, especially when they require some lead time before yielding benefits (e.g., see discussion of Options 2.3 through 2.6).

Otherwise, Congress might wait until more information becomes available concerning the likely outcomes of an action (compared to the status quo) in order to avoid precipitous action and legislation that may not have the desired long-term effect or that may quickly be outdated by changes in technology. Delaying action will also allow Congress to have the benefit of additional information about the course of case law (on software interfaces, patent litigation and appeals, etc.) and on the impact on software developers and users of the current legal environment and uncertainties. In the nearer term, Congress could initiate strategic information- and input-gathering processes (e.g., Options 1.5 and 3.3, see also section below on “planning for the future”) and use these as a basis for evaluating

[103] See discussion, ibid., p. 550

[104] CONTU noted language in the House and Senate reports accompanying the 1976 Copyright Act indicating that the act did not need to be amended to include computer programs, and language indicating that computer programs were copyrightable as “literary works. ” (See *CONTU Report,* op. cit., footnote 16, p. 16.)

[105] Hersey dissent, *CONTU Report,* Op. cit., footnote 16, p. 31.

[106] See, e.g., *E.F. Johnson v. Uniden,* 623 F. Supp. 1485, 1498.

[107] See, e.g., Morton David Goldberg and John F. Burleigh, op. cit., footnote 76, p. 294.

longer-term options.[108] Such a strategy fight be especially helpful in determining whether statutory changes are necessary and, if so, what definitional problems and uncertainties will be addressed. Possible disadvantages of waiting are that incremental accommodations through the case law may conflict over time, as the case law continues to evolve. As is the case with current legal uncertainties, the uncertainties that ensue will affect smaller/poorer firms and individuals (that do not have the resources to ‘‘ride it out) more than large firms with deep pockets.

If Congress chooses to take action, it faces a third choice--how comprehensively to act. Congressional actions could take the form of: measures to address ongoing institutional problems (e.g., Option 2.4); measures to seek ‘‘cooperative, ’ rather than legislative, clarification of uncertainties (e.g., Option 1.5); or legislative measures to amend current copyright and patent statutes (e.g., Option 1.1), or create *sui generis* protection (e.g., Option 1,4). Depending on the specific action or actions taken, the overall effect might:

1. explicitly affirm the status quo and course of case law (e.g., in terms of the scope of copyright and patent);
2. make small adjustments at the margins of copyright and patent (e.g. through procedural changes);
3. clarify or modify the scope of patent and/or copyright (e.g., through definitional changes), but leave the basic paradigms the same;
4. introduce one or more *complementary, sui generis* regimes tailored specifically to certain aspects of programs and software functionality, recognizing particular patterns of innovation; or
5. develop a *sui generis* regime for software to *substitute for* copyright and/or patent protection, tailored to encompass program code as well as software design and functionality.

These alternatives appear to impact increasingly on the present intellectual property system as the list progresses. But it is not necessarily the case that the least disruptive choices (e.g., explicitly afirming the status quo) can, or should, be selected more quickly than the others. Any of these must result from careful deliberation and crafting, in order to specify clearly and unambiguously what is and is not covered, and what exceptions, if any, are to be made (e.g., along the lines of reverse engineering, fair use, etc.).[109] The following sections discuss the policy issue areas in the context of the above choices for congressional action.

Policy Area 1: Difficulties that the functional aspects of computer programs present in determining the appropriate scope of copyright protection

The functional aspects of computer programs pose difficult questions for application of the copyright law, most notably the appropriate scope of copyright. The traditionally ‘‘fuzzy” line between idea and expression in copyright law is complicated by the need to determine an appropriate scope of protection in order not to cover for the program procedure, process, system, method of operation, or concept, contrary to the intent of the current law (17 U.S.C. 102(b)).

Despite the advantages of incremental accommodation within the copyright law, there may be a point where it becomes preferable to augment or complement the existing framework rather than extend the scope of copyright to fit software-perhaps, cumulatively, at the expense of other types of works. Sometimes, what is in question is the extent to which copyright (perhaps, in concert with trade secret law) is to be interpreted to protect the functional and design aspects of the program in addition to the code. This is the essence of the current debate over application of copyright to features of a program’s external design and/or user interface, and in the debate over reverse engineering and recompilation. If or when Congress decides to take action, options include the following:

Definitional Issues and the Scope of Software Copyright

To clarify the scope of copyright protection for software beyond the code, Congress might want to explicitly include or exclude one or more aspects of software, such as computer languages, algorithms, design specifications, user and other interfaces. If it

[108] Under this strategy, Congress might also conduct Series of hearings on the issues and assess the results of the executive branch review of the PTO.

[109] For discussion of how this was accomplished for the SCPA, within a framework for evaluating proposed changes to the intellectual property system, see Kastenmeier and Remington, op. cit., footnote 30, esp. pp. 438451.

chose to explicitly include one or more of these, Congress could:

Option 1.1a: Expand upon the **Copyright Law's current language on** "subject matter of copyright" in Section 102 to specify that [computer languages/algorithms/design specifications/user and other interfaces/. ..] are copyrightable subject matter.

Alternatively, if it chose to make explicit exclusions, Congress could:

Option l.lb: Expand upon the Copyright Law's current language on "subject matter of copyright" in Section 102 to specify that [computer languages/algorithms/design specifications/user and other interfaces/. . .] are not copyrightable subject matter.

In order to do this, statutory definitions for software-specific terms like ‘‘computer language,' ‘‘algorithm, ' etc. would have to be developed for Section 101, along with means to keep them current or update them as technological changes require. This would not be easy, and would require input from the technical, as well as legal, communities to ensure that the resulting language is unambiguous.

If (under Option l.lb) Congress chooses to explicitly exempt any of these from inclusion within the scope of copyright, then it must determine whether they are to be left to the public domain, trade secret law, patents, or to new, *sui generis* laws (see Options 1.3 and 1.4 below).

Alternatives to Current Treatment of Programs as Literary Works

Option 1.2: Establish a separate category in the Copyright Act for "computer programs," instead of treating them as literary works.

In the international arena, however, this is counter to current U.S. standards of "adequate' protection for software as or akin to a literary work. Also, there would be a period of uncertainty as a new body of case law developed. An advantage of this approach would be that the courts would not have to apply the same principles to software and other literary works, whose economics, patterns of innovation/dissemination, and useful life spans are quite different. Using this approach, though, the term of protection would remain the same as for other copyrighted "literary and artistic' works.

Instead of establishing a separate category for software within Section 102(a)), Congress might limit the scope of "literary" copyright to the code, with the possibility of adopting a *complementary* regime for elements of software design and functionality.

Option 1.3: Leave "computer programs)' within the category of literary works but establish legislative bounds holding the extent of copyright protection as a literary work to the code (as text), not to the behavior of the program when it is executing or to "interfaces.'> Determine whether the latter are to be covered by a complementary, *sui generis* regime.

This option would continue to allow copyright protection for the program code both in the United States and in other countries, under the provisions of Berne. However, it would leave room for a different mode of protection for elements of design and functionality, such as the program's external design and the design of user interfaces. Alternatively, these could be left to the public domain except as protected by patent and/or trade secret law.

At this time, it is not clear to what extent the courts will find these elements to be protected by copyright law. Protecting them explicitly through a new, *sui generis* law would reduce uncertainty about their protection, and provide for features not permitted under copyright, such as a shorter term length (but long enough to allow some reasonable lead time), or compulsory licensing to facilitate standardization. A *sui generis* law might also have software-specific criteria for infringement or for exemptions and could impose a different threshold standard for innovation than copyright's ‘‘originality' criterion.

A disadvantage of this approach, as for other options, is that it may be difficult to define what is and what is not covered under copyright and under the new, *sui generis* law. A new body of case law would have to develop, as would international agreements, particularly regarding the *sui generis* mode of protection.

Another option, which represents a more significant change from the existing modes of protection for software, is to *replace* copyright protection for software with a new form of protection tailored especially for software.

Option 1.4: Withhold copyright protection from “computer programs” and substitute protection under a *sui generis* framework, including protection for the program code, as well as other elements of program functionality and design.

This approach would replace copyright protection for software with a *sui generis* regime tailored to the protection of computer programs. The new law could address both issues of scope and of reverse engineering.[110] It could either explicitly include or exclude ‘‘interfaces and could determine under what circumstances, if ever, reverse engineering was permissible. A different length of protection could reflect possible differences in market life or purchase patterns between computer programs and traditionally copyrightable works. The availability of protection tailored especially for software might also decrease use of patent protection for some software-related inventions.

As with the *sui generis* choice in Option 1.3, it may be difficult to define what is and is not covered under the new law. Another disadvantage of a *sui generis* law designed to substitute for copyright is that there would be much uncertainty during a transitional period, as the case law develops. In addition, as with other sui generis options, there would be no established international treaty structure.

Reverse Engineering

The issues and uncertainties concerning reverse engineering might be handled by clarifying or modifying the scope and subject matter of copyright as described in Option 1.1 or Option 1.4 above. Another alternative would be to clarify the existing statute concerning ‘‘fair use’ (Section 107) and existing limitations on exclusive rights in computer programs (Section 117). This clarification could be accomplished through cooperation or through legislation. *A "cooperative approach could either stand alone or be a precursor to legislation. This type* of approach has been used in the past to reduce uncertainties about the acceptability of certain photocopying practices.[111]

Option 1.5: Direct the Copyright Office, with the assistance of software producers, software consumers, educators, and representatives of the public at large, to develop practical guidelines regarding “fair use” of programs and “essential steps in the utilization” of programs. These guidelines should address what reverse engineering practices, if any, are permissible.

If Congress decides to pursue a legislative approach to deal with the uncertainties surrounding reverse engineering, it might:

Option 1.6: Establish legislative guidelines regarding “fair use” of computer programs and Section 117, especially with respect to fair use for unpublished works (source code as trade secret, object code “published’ and reverse engineering.

Among other things, these measures would establish whether incidental copies made during the course of reverse engineering are or are not copyright infringements and/or the extent to which factors such as the purpose of reverse engineering, whether or not a resulting program has taken protected expression from the first, etc. should be taken into account. Legislation might develop from study and input from the industry and the public, whether conducted under Option 1.5 orotherwise.[112]

Policy Area 2: Difficulties in determining the scope of patent protection for software-related inventions and algorithms and the considerable technological and institutional challenges the U.S. Patent and Trademark Office faces in examining applications in these areas of art

[110] For example, the new law could have a shorter span of protection than copyright, a higher standard of originality, compulsory *licenses, and special* provisions for reverse engineering.

[111] CONTU recommended that the Register of Copyrights and others periodically study and report on photocopying practices in and out of libraries. Based on lengthy consultations with concerned parties, CONTU developed guidelines for library photocopying that were intended to be fair and workable. (*CONTU Report,* op. cit., footnote 16, ch. 4, See also 17 U. S. C. A., sec. 108, pp. 136-137.)

See also footnote 119 and accompanying discussion.

[112] For example, Section 108 of the Copyright Act provides detailed guidance for library and archival photocopying; in developing the language for that section, the conferees agreed that CONTU’s guidelines were a reasonable interpretation of sec. 108(g)(2). (17 U. S. C. A., p. 136.)

Whether and to what extent software-related inventions are the subject of utility patent protection has been an issue before the courts and the PTO since the early 1960s. The U.S. Supreme Court has examined the issue of patentability of software on a number of occasions *(Gottschalk v. Benson, Parker v. Flook,* and *Diamond v. Diehr),* attempting to delineate the limits of patentable subject matter. The PTO faces considerable challenges in examining applications involving computer processes. If or when Congress wishes to take action, options include the following:

Statutory Subject Matter

Under current interpretations of patent law, patents may be granted for certain parts of a program's function. The same program may embody many patentable inventions, or none at all, depending on which parts of the program function are novel, nonobvious, and meet the requirements for statutory subject matter.

The case law and PTO guidelines indicate that patents may not be granted for a program function called a' 'mathematical algorithm' (see footnote 20, supra). The definition of 'mathematical algorithm' has had considerable discussion; currently it seems to refer to a program function that is a "mere calculation. " According to PTO guidelines, claims that include calculations expressed in mathematical symbols contain a mathematical algorithm.[113] On the other hand, the function is not considered ''mathematical' if it can be stated in terms of its operations on things in the 'real world. [114] Over the past decade, patents have been issued for software-related inventions such as linear-programming algorithms, spell-checking routines, logic-ordering operations for spreadsheet programs, brokerage cash-management systems, and bank college-savings systems. To some industry observers, there appears to be varianc--or, at least, uncertainty on their part-in how PTO guidelines are being applied during examination.115

To reduce uncertainties and clarify legislative intent, Congress could explicitly address the question of patentability for software-related inventions and for certain algorithms. However, either of the options below would face even more difficult definitional problems than those of the copyright options. For example, a good deal of the software debate has focused on whether PTO should grant "software patents. ' The term "software patent" does not correspond to any PTO category (see footnotes 19, 89 and 90, supra). As it is used in the debate, 'software patent' appears to refer to patents that can be infringed by a computer program executing on a general-purpose computer.[116] However, this class of inventions includes more than just ''software patents. It also includes, for example, traditionally patentable processes which happen to employ a computer.117

The need to make the distinction between nonstatutory "mathematical algorithms' and statutory inventions results from the courts' efforts to interpret, in the context of software-related inventions, the patent doctrine that "laws of nature" are not statutory subject matter. Any effort to redraw the line between statutory and nonstatutory software-related inventions is likely to encounter serious definitional problems.

In addition, the types of processes and apparatuses that typically are the subject of "software patents" can be claimed in a way that covers both hardware and software implementations. At present, the form of implementation (hardware or software) does not deternin e whether an invention is statutory subject

[113] U,S, *patent* and Trademark Office, *Computer Programs and Mathematical Algorithms,* September 1989, p. *8.* However, the PTO guidelines state that a' 'mathematical algorithm" may be present in prose form, without the presence of mathematical symbols: ''It is not always possible to determine by inspection of the claim whether it indirectly recites a mathematical algorithm; in such instances the analysis 'requires careful interpretation of the claim in the light of its supporting disclosure. ' *Johnson,* 589 F.2d at 1079, 200 USPQ at 208. " Ibid. Despite the presence of a mathematical algorithm, the claim may be statutory. See discussion of the ''two-part test'' in ch. 4 and in pp. 5-12 of U.S. Patent and Trademark Office, *Official Gazette,* Aug. 9, 1989.

[114] *In re Bradley,* 600 F.2d 812 (C. C.P.A. 1979). Inventions that process architectural symbols (*In re Phillips,* 608 F.2d *879 (C.* C.P.A. 1979)) or translate languages (*In re Toma,* 575 F.2d 872 (C.C.P.A. 1978)) were not found to be "mathematical."

[115] See footnote 85, supra (ADAPSO, op. cit.).

[116] Inventions of this type belong t. a genre of inventions PTO refers to as ''computer-related inventions' or 'computer-implemented process patents" (see footnotes 86 and 90, supra). In the request for comments for the Advisory Commission on Patent Law Reform, PTO also used the term, "computer program-related inventions" *(Federal Register, vol. 56, No. 95,* May 16, 1991, p. 22702-22703).

[117] An example would be the rubber-curing process found to be statutory by the Us. Supreme Court in *Diamond* v. *Diehr.* There does not appear to be much public concern about these types of inventions.

matter. The only issue is whether the inventor is attempting to claim a ‘‘mathematical algorithm. ” However, if software implementations were not statutory subject matter or could not infringe a hardware implementation, then some hardware-based inventions could have their value appropriated by software implementations.

Option 2.1: Refine the statutory definition of patentable subject matter to provide guidance for the courts and PTO. Legislation might address the extent to which processes implemented in software or “mathematical algorithms” are or are not statutory subject matter. Legislation might also address the issue of special exemptions, such as for research and education.

Option 2.2: Exclude software-related inventions and/or algorithms from the patent law and create a special, *sui generis* protection within a patent framework for some inventions. The latter might have a shorter term, lower criteria for inventiveness, and/or special exemptions from infringement.

Other measures could address some of the challenges facing PTO regarding the database of prior art and the timeliness and quality of examination:

Prior Art and Examination Quality and Timeliness

PTO has been grappling with institutional problems such as examiner training and turnover, length of pendency periods for patent applications from filing to issuance, a backlog of applications, and the quality and extent of the prior art database.[118] OTA views these problems as serious since they may affect the quality of the patents issued and create additional burdens for software developers and users (e.g., “landmine” patents—see box l-C). A major problem for patent-system administration with respect to software-related inventions and algorithms is the incomplete stock of ‘‘prior art’ available to patent examiners in evaluating patent applications for processes involving computers, especially those involving software and algorithms.

“Filling in” the prior art database (with both patent and nonpatent prior art) is important for improving the quality of examination. Improving electronic search and retrieval capabilities for PTO’S own database is also critical, because it is used by PTO examiners during the application process and is also used by the public. The public’s access to an adequate prior-art database is crucial because it allows software developers to review the status of the art and to determine what has already been covered by patent. Given this information, developers can make more informed decisions about their design alternatives, their choice of patent protection versus trade secret, and what might be worthwhile areas for further research and development.

Measures to address the quality of the prior-art database and examination (e.g., Options 2.3 through 2.6) will benefit PTO and the public, no matter what other legislative options are chosen. Because they require some lead time, Congress might select a strategy of initiating one or more of these options now:

Option 2.3: Encourage establishment of a supplementary repository of nonpatent prior art, either public or private.

Patent examiners for the most part are limited to prior art that is already the subject of patent protection. Unlike other areas of technology, software prior art consists in large part of software products that are not a part of the PTO prior-art database. Such a supplemental prior-art database would expand the background against which examiners would compare patent applications, and would allow the software community to inform the PTO of art which is in the public domain but of which the PTO would not be aware because it is not patented. This would prevent the patenting of art which is arguably “old.”

Whether publicly or privately developed and maintained, a supplemental repository of prior art would allow members of the software community to participate in upgrading the bank of prior art for software and thus assist in maintainingg the quality and legitimacy of specific, issued patents for software-related inventions. Public access to such a database might also be encouraged, allowing developers to track innovation and make decisions about future

[118] The Department of Commerce has established an Advisory Commission on Patent Law Reform, due to report in 1992. The commission is examining 13 sets of issues regarding the patent system; these include protection of what PTO terms “computer-related inventions, ” as well as procedural matters such as a first to file system, automatic publication of applications, and the term of patent protection.

research and development and about intellectual property protection for its products.

One nongovernmental response to these needs is the effort to establish a nonprofit Software Patent Institute to provide a supplementary repository of prior art for PTO and others and to offer educational and training opportunities for PTO staff.[119] Congress might wish to monitor the progress of this effort in order to determine what role, if any, government should play.

Option 2.4: Encourage PTO to include in its ongoing automation program means for improved data retrieval, such as flagging, cross-indexing, etc. Encourage the PTO to use this improved database and increased access to the prior art to monitor activity and trends in "hot" areas of art, as well as anticipate and plan for changes in staffing and expertise.

Examiners make comparisons between the invention described in the patent application and the prior art in order to make determinations about the novelty and nonobviousness of the invention described in the patent application. Including the means to cross-reference patents among different areas of art would give examiners easier and better access to pertinent prior art. Increased awareness of the nature of patents for software-related inventions issued by PTO and increased access to those patents by examiners would improve the quality of the examination with respect to nonobviousness and novelty.

In addition to these benefits, this system would enable the PTO to review the trends in prior art both generally and within specific areas of art to determine staffing needs (numbers and skills of examiners). Matching examiner staffing levels and expertise to the changing quantity and character of the influx of applications could improve the quality and efficiency of the examination process and, as a result, yield a higher quality of issued patents. Such a system would also improve PTO'S ability to respond to questions generated for congressional oversight.

Option 2.5: Encourage PTO'S ongoing efforts to improve funding, training, and support for PTO examiners, in order to assure high-quality examinations. Examination quality depends both on the clarity of the examination guidelines and on training and support for examiners.

The PTO has cited high turnover of examiners—particularly in the computer-related art units-as a concern. Well-trained examiners familiar with the prior art and the examination guidelines are important to the quality of issued patents. Steps are needed to induce or enable highly trained examiners in the computer arts to stay within the PTO, and avoid high turnover and the expense and delay of training new hires.

Option 2.6: Encourage PTO to continue to seek input from the software community in formulating examination guidelines, developing classification systems, anticipating technical change, improving the prior art database, and determining appropriate examiner qualifications. Expand efforts to communicate PTO practices and guidelines to the software community, especially in the period following new Court of Appeals for the Federal Circuit cases.

PTO/software community communication could be improved by establishing an external computer science and software engineering advisory committee for IWO, with balanced representation. PTO might also seek technical review from experts in government (e.g., at the National Institute of Standards and Technology) in reviewing changes to examination guidelines.

A procedure for challenging software-related patents (presenting additional prior art to the PTO) on an expedited basis could aid in the development of the prior-art database. Challenges could take place prior to issuance of a patent, or shortly thereafter, and could help ensure that patents not be issued for developments that are in fact well-known or well-established ‘‘inventions. ’

Option 2.7: Because of gaps in the prior art, either: expedite challenges to newly issued patents in the software area or establish procedures for preissuance challenges.

Publication of applications after 18 months, whether or not a patent issues, would make the subject matter available to the public. If the subject of the application is deemed patentable by the PTO,

[119] See ch. 2 discussion of efforts to establish a Software Patent Institute.

the published information becomes proprietary. If not, and the application is not withdrawn, the information becomes part of the prior art. The major advantage of this approach is the avoidance of "landmine" patents that issue after years of pendency. This system of publication would allow developers to avoid investing in research and development on technology that is already the subject of an application and, potentially, covered by a patent.

Option 2.8: Because of application backlog and pendency problems, and the possibly shorter market life of software, publish applications after 18 months, whether or not issued, or provide PTO with resources to shorten pendency for software-related patent applications to 18 months.

Another consequence of this system could be a reduction in the number of applications for patents, as some developers chose to avoid the risk of losing trade secret protection for their inventions. If the pendency for patent applications could be shortened, such an approach would likely be unnecessary, as the problem of "landmine" patents would be reduced by timely issuance of patents.

Policy Area 3: Complications facing libraries and other developers and users of digital information, **especially mixed media** works from several different categories of copyrightable works

Decisions affecting intellectual property and software can also affect digital information and the industries, individuals, and institutions that create and use it. Government may have a role in clarifying "fair use' with regard to digital information. Guidelines might be developed to clarify the rights of libraries to make archival copies of digital information or to provide copying and other services to patrons (computer networks allow patrons to be people miles away, not just those within library walls). The rights of libraries to lend, archive or share traditional materials have been well established in the copyright law, but in the case of digital information are often defined by contracts with information providers.

Clarification might also be provided on the extent to which computer-based catalogs can be enhanced by incorporating material from underlying works. While such clarification might be made through legislation, several sets of nonstatutory fair-use guidelines exist,[120] and it may be useful to update them or to develop additional ones through consultation with users and other interested parties. Publishers rely on contracts specifying what users may do with data to deal with the uncertainties about what users' rights are to ''download, ' ''use, ' or 'copy' data under fair-use principles of the copyright law. Clarification of users' rights under copyright could simplify or reduce the need for such contracts. Guidance would, however, have to be carefully crafted for a wide variety of users, products, and technologies. If or when Congress decides to take action, options include the following:

Use of Works in Electronic Form

Option 3.1: Clarify "fair use" guidelines with regard to lending, resource sharing, interlibrary loan, archival and preservation copying, copying for patron use, for works in electronic form.

Option 3.2: Establish legislative guidance regarding "fair use" of works in electronic form and what constitutes "copying" and "reading" or ''using. "

These clarifications would reassert Congress's intention, as expressed in copyright law, to establish limitations on the rights of copyright holders and to permit certain uses of information for research and educational purposes. *Alternatively, a ' 'cooperative' alternative that might or might not lead to subsequent legislation would be to:*

Option 3.3: Direct the Copyright Office, with the assistance of producers and users of electronic information, to develop and disseminate practical guidelines regarding "fair use" of works in electronic form and what constitutes "copying" and "reading" or "using."

Whether established through legislation or through nonstatutory, cooperative guidelines, these clarifications would require careful crafting, with

[120] Guidelines for fair use related to educational and nonprofit Org animations were incorporated into the House Committee Report prior to the enactment of the 1976 Copyright Act. Another set concerning off-air taping of broadcast television was approved by the House Subcommittee on Courts, Civil Liberties and the Administration of Justice (Congressional Reeord Sec. E?4751, Oct. 17, 198 1).

input from all interested parties—users, as well as producers-in order to cover the wide variety of users, products, and technologies existing now and in the future.

Multimedia Works

Multimedia works raise questions in two areas. First, what type of protection should they be afforded-as single works or as collections of different works in different categories. The second deals with the incorporation of copyrighted works in a mixed media work. Guidelines may be needed to determine what rights should be obtained, for example, in determining whether a multimedia presentation on a personal computer constitutes a public performance or merely an adaptation of the music or drama incorporated within.

option 3.4: Clarify the status of mixed media works with regard to their copyright protection.

Permissions and Royalty Collollection

The difficulty of obtaining permission for including images, text, or other copyrighted works incorporated into multimedia databases could be eased by the creation of a royalty collecting agency or clearinghouse (or perhaps several agencies for different types of works). The ability to deal with one source, or a small number of sources, for permission to use these works would aid development of multimedia projects, as would creation of standards and conventions about the royalties to be paid.

Option 3.5: Create, or encourage private efforts to form, clearing and royalty collection agencies for groups of copyright owners.

The responsibility for creating such agencies probably rests with the rights holders, rather than with government. There may, however, be a role for government in easing antitrust or other regulations to encourage the creation of such organizations.

Preparing for the Future

Each of the principal policy areas and intellectual property issues discussed in this report is complicated by the complexity of software and computer technologies and by the rapid pace of change in these technologies. Congress and the courts could begin to benefit now from institutional means to understand and remain current about the emerging issues surrounding them.

Establishing a "Congressional Commission on Computing Technologies" would help. The commission (composed of technology experts from academia and the private and public sectors) could be charged with monitoring the fields of computer and software technologies and reporting periodically (perhaps each Congress) on the status of, new directions in, and problems facing these fields. The commission reports could be helpful to Congress in anticipating future areas of policy concern and topics for further study.

Courts are asked to make difficult decisions about technology that is new, changing, and complex. *Congress might consider measures that the courts would find helpful as they deal with complex cases involving computer and software technologies.* Such measures might include special software and computer technology courses tailored for the judiciary, use of special masters, and other means for educating the courts and keeping them abreast of developments in hardware/software technologies. The largely nontechnical judicial staff-judges and clerks—could, in this way, be tutored about current computer/ software technologies, thus contributing to a high standard of judicial decisionmaking. Special masters, focused in the area of computer litigation, could provide understanding and expertise on a continuing basis, and make technical determinations when needed.

For the longer term, Congress might also explore the merits of establishing a ''faster track" for intellectual property litigation concerning software and software-related inventions. Software products have a relatively brief market life. Courts, already burdened by caseloads and crowded dockets must render decisions for a fast changing and complex industry. An expedited time period for software-related litigation would assist the courts in reaching timely decisions.[121]

[121] Congress has in the past made special provisions to shorten the time period for litigation for other patented products; the Waxman-Hatch Act provides for a 30-month period for litigation of certain actions for patent infringement involving pharmaceuticals and certain drug and veterinary biological products which must undergo an often lengthy approval process with the Food and Drug Administration. (See 35 U.S.C. 271 and 21 U.S.C. 355(j) (4)(B)(iii).)