Chapter 3

The Intellectual Property Bargain and Software

In the United States, an *“intellectual property bargain”* underlies the concept of intellectual-property protection. This “bargain” between creators and society balances two social objectives: 1) it encourages the production and dissemination of new works and inventions (by providing economic incentives to creators), and 2) it promotes access to and use of these works and inventions.[1]Thus, the limited monopoly granted to authors by copyright and to inventors by patents is a quid-pro-quo arrangement to serve the public interest, rather than a system established primarily to guarantee income to creators. (See app. A for reviews of copyright, patent, and trade secret protections as they pertain to software.)

COPYRIGHT

Copyright is granted to authors for the creation of certain classes of works.[2]The economic underpinnings of copyright assume that to profit from a work, the author will publish or otherwise disseminate it to the public.[3]Copyrights, which are relatively easy to obtain and long-lasting compared to patents, are intended only to protect the ***expression*** in a work from unauthorized copying, not to protect the underlying ideas or functionality from use. Even “expression” is not protected from independent creation.

The recommendation by the National Commission on New Technological Uses of Copyrighted Works (CONTU) that copyright protection be explicitly extended to all forms of computer programs was established in the 1980 amendments to the Copyright Act.[4]Even then, CONTU recognized certain difficulties in applying copyright (which does not protect ideas, processes, or procedures) to software, which is inherently functional. A particular concern was the impossibility of establishing a precise line between the copyrightable “expression” in a program and the noncopyrightable processes it implements—the distinction between “expression” and “idea.”5 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.[c]CONTU concluded, however, that these questions should be answered on a case-by-case basis by the Federal courts.[7]Many continue to believe that traditional copyright principles should continue to be applied to software, because difficulties in distinguishing between idea and expression are not unique to software and because copyright law has been able to embrace many new forms of authorship within existing principles.[8]

The 1986 OTA report, *Intellectual Property Rights in an Age of Electronics and Information,* discussed the increasing difficulties of applying copyright to *functional works* such as programs.[9] Some of these difficulties are shown today by ongoing “look and feel”1° and “structure, sequence,

[1] **See** U.S. Constitution, art. I, **sec.** 8, cl. 8. For a discussion of this bargain and the public interest in intellectual property protection,**see U.S. Congress, Office** of Technology Assessment, Intellectual ***Property Rights in an Age of Electronics and Information,*** **OTA-CIT-302;** and ***Copyright and Home Copying: Technology Challenges the Law,*** **OTA-CIT-422** (Washington, DC: U.S. Government Printing Office, October 1989), **ch.** 3.

[2] Title 17, **U.S.C.** 102(a).

[3] See **OTA-CIT-302,** op. cit., foomote 1, p. 7 and **ch.** 6. Copyright inheres in a work as soon as it is created and also exists for unpublished works.

[4] **CONTU** recommended that programs be protected as literary works. **CONTU's** definition of ● ’computer program” was added to Sec. 101 of the **Copyright Act** of **1976 and** a new Sec. 117 was added limiting computer-program copyright holders’ exclusive rights.

[5] **See Peter** s. **Menell, “An Analysis** of **the Scope** of Copyright Protection for Application**Programs,”** ***Stanford Law Review,*** **vol. 41, 1989, p. 1047**.

6“Fi~ **Report of the National Commission 011 New** Technological Uses of Copyrighted Works,” **July** 31.1978. **pp. 22-23.**

[7] Ibid., p. 23.

[8] For **discussion of this view, see Morton** David **Goldberg and John F. Burleigh, “Copyright Protection for Computer Programs,”** ***AIPLA Quarterly Journal, vol.*** 17, No. 3, 1989, pp. 294-322. Goldberg and **Burleigh** argue that the courts have (as Congress intended) conscientiously applied traditional **copyright principles to software cases and, for the most part, are reaching** prep= and well-feasoned results (ibid., p. 296).

[9] **See OTA-CIT-302, op. cit., footnote 1, pp. 78-85. The 1986 report identified three types** of **copyrightable** works: **works** of **art, created for their own** ***intrinsic value; works*** of fact, such as databases, whose value lies in an accurate representation of reality; and works of function, such as computer **programs,** which **use information** to describe or implement a process, procedure, or algorithm.

[10] **“Look” is often taken to mean the appearance of screen displays,** “feel”to mean the way the program**responds when the user selects options or enters** commands. User **interfaces, including** graphic icons or combinations of keystrokes to represent functions like “save” or “delete,” are part of “look and **feel.”**

and organization" copyright suits. Moreover, market changes, like the almost-hundredfold increase in PC use since CONTU, make the financial stakes much higher.

PATENT

Copyright protects the expression of an idea. Patent protects the ***technological application*** of an idea in a machine or process.[11] A patent precludes "practice" of the invention (e.g., making, using, or selling the claimed invention) by others, even if they invent it independently. But the requirement for patentability is stringent: the invention must be useful, novel, and nonobvious compared to prior discoveries (the "prior art") that are patented, in the public domain, or otherwise widely known.[12] While publication is not required for copyright, patent is granted in exchange for full *disclosure* of what the inventor considers the best way of implementing or practicing the invention.[13] The purpose of the patent is to "teach" others and thereby stimulate technological progress as they seek to build on (or invent around) the discovery.

The availability of patent protection for software-related inventions was unclear (generally considered "not applicable") until the early 1980s.[14] Since 1981, there has been renewed interest in patents for software-related inventions.[15] Over the past 7 years, 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.[16]

In the last year, some patent lawsuits concerning software-related inventions and controversies concerning patents for algorithms have become highly visible. These lawsuits and specific controversies have focused *concerns* over the appropriateness of patent protection for software-related inventions and algorithms. These concerns arise both from lack of belief that patents in computer-program processes encourage technological progress, as well as from the practical problems that software-related inventions and algorithms raise for patent-system administration.

One of these problems is the incomplete stock of "prior art" available to patent examiners in evaluating patent applications for processes involving computers, especially those involving software and

[11] The statutory subject matter of a 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.

For an overview of patents, including a discussion of criteria for patentability and how a patent is obtained, see 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), ch.3.

[12] Although all "original" programs are generally eligible for copyright, the fractionof programs potentially able toqualify for patent protection is much smaller. For one thing, the U.S. Patent and Trademark Office (pTO) position is that computer programs per se are not patentable, as opposed to patentable computer processes and algorithms (see footnote 15 below).

In the early 1980s, some commentators estimated that over 90 percent of computer-program inventions would not in principle meet the patent requirement that the invention benonobvious, compared to the prior art. Therefore, they estimated that patent protection would only be relevant to about 1 percent of all software. (Findings of ABA Proprietary Rights in Software *Committee* (1983), cited in Cary H. Sherman, Hamish R. Sandison, and Marc D. Guren, *Computer Software Protection La-w* (Washington, DC: The Bureau of National Affairs, Inc., 1989), pp. 401-408 and note 41.) (OTA NOTE: The 90 percent and 1 percent figures do not refer to the percentage of patent applications that result in a patent being issued.)

[13] The specification disclosed illustrates one implementation of the invention; others may be possible. The patent application must describe the invention adequately to allow a person of "ordinary" skill (in the particular area of technology) to make and use the invention.

[14] In 1972, the Supreme Court stated *that* certain inventions performed by computers could be patentable subject matte(*Gottschalk v. Benson,* 409 U.S. 63 (1972)), A 1981 Supreme Court decision (*Diamond v. Diehr, 450* U.S. 175 (1981)) helped clear the way for patent protection for some software-related inventions by clarifying the circumstances under which inventions performed by computers could be patentable subject matter.

[15] The Supreme Court has not ruled as to whether computer programs per se constitute patentable subject matter. 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 man-made mathematical algorithms and mathematical algorithms representing discoveries of scientific principles and laws of nature (which have never been statutory subject matter). For a process claim involving a mathematical algorithm to be patentable, the claim excluding the algorithm is required to be statutory subject matter—i.e., the claim must be for a process, machine, etc. Trivial post-solution activity like displaying a number is not sufficient. ("Patentable Subject Matter: Mathematical Algorithms and Computer Programs," 1106 O.G. 4, Sept. 5, 1989; also contained *in Patent Protection for Computer Software: The New Safeguard,* Michael S, Keplinger and Ronald S. Laurie (eds.) (Englewood Cliffs, NJ: Prentice Hall Law and Business, 1989), pp. 9-42.)

[16] In this paper, OTA sometimes uses phrases like "patents for software-related inventions, ""soflvvare-related patents," or "patenting algorithms" to refer generally to patent protection for computer-implemented processes and algorithms. The U.S. Patent and Trademark Office (PTO) considers terms like "softwarepatents" to be a misnomer because they may be interpreted to mean that a computer programmer se (i.e., the sequence of coded instructions itself) is patentable, as opposed to the underlying computer process it carries out. (M. Keplinger, G. Goldberg, and L. Skillington, PTO, comments on draft paper, Dec. 18, 1989, pp. 1-2.)

algorithms.[17] The published literature does not completely embody the development of the fields of software and computer science. In many cases, important prior art exists only in product form and is not described in print form such as articles in technical journals.[18] Another problem is the lack of special classifications or cross-references to issued patents. As a result, it is virtually impossible to find, let alone count or profile, all software-related or algorithmic patents. This means that patent examiners and the public have no effective way of searching and studying such patents.

Another problem is the long time lag between patent application and issuance, compared to quick-moving software life cycles. Someone may develop and bring a software package to market, unaware that it will infringe on a patent applied for by another developer, but not yet granted. These are called "landmine patents," and can occur in other areas of technology besides software.[19]

TRADE SECRET

Trade secret protection, provided under individual State laws, protects against use or willful disclosure of the secret by others (but not against independent discovery). Most foreign nations outside of Western Europe do not have extensive trade-secret laws. However, most developed countries do have some form of legal protection for confidential business information[20] and contracts or licenses can often provide equivalent protection abroad.

Trade-secret information maintains its status so long as the information is not publicly disclosed.[21] Unlike copyright or patent, there is no limitation on its duration. Trade secret has been the favorite mechanism to protect mainframe and minicomputer software and its underlying ideas, logic, and structure because programs are licensed to specific customers, not the mass market. Mass-marketed PC software is sometimes released with "shrink wrap" licenses intended to maintain trade-secret status (see app. A). Software that is protected effectively as a trade secret does not become prior art. This can adversely affect patent examinations and lead to "reinventing the wheel."

[17] Interest in patenting software-related inventions and algorithms is relatively new. Copyrighted software deposited athe Copyright Office is not readily searchable for patent purposes. Also, trade secrecy has been a major form of software protection, and trade-secret information may not constitute part of the prior art, Therefore, the prior art readily searchable by patent examiners and the public has gaps. This potentially allows patents to issue for computer-process inventions that are already known in the industry or that represent only minor improvements. However, the PTO is working to improve the file of prior art for search purposes.

[18] Ronald Laurie, Irell & Manella, personal communication, Dec. 21, 1989.

[19] However, some observers believe that timing poses special problems for software-related inventions because of a combination of factors 1) the decentralized nature of the software industry, 2) difficulties in determining the prior art, and 3) the rapid rate of software-product development and short product life cycles, compared to the time required for processing a patent application. (Brian Kahin, Harvard University, personal communication, Dec. 1, 1989; and Brian Kahin, "The Case Against 'Software Patents'," personal communication on Dec. 1, 1989.)

[20] Michael Keplinger, Gerald Goldberg, and Lee Skillington, Patent and Trademark Office, personal communication, Dec. 18, 1989.

[21] To maintain trade-secret protection for software, developers may require that employees or transferees hold the information in confidence.