1992

Copyright law as impacted by changes in computer software technologies

Richard O. Ward
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

---

Recommended Citation

Ward, Richard O., "Copyright law as impacted by changes in computer software technologies" (1992). *Master's Theses*. 432.

DOI: <https://doi.org/10.31979/etd.5p85-zxvp>
https://scholarworks.sjsu.edu/etd_theses/432

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313 761-4700 800 521-0600

Order Number 1350126

**Copyright law as impacted by changes in computer software
technologies**

Ward, Richard O., M.L.S.

San Jose State University, 1992

Copyright ©1992 by Ward, Richard O. All rights reserved.

U·M·I

**300 N. Zeeb Rd.
Ann Arbor, MI 48106**

COPYRIGHT LAW AS IMPACTED
BY CHANGES IN COMPUTER
SOFTWARE TECHNOLOGIES

A Thesis

Presented to

the Faculty of the Division of Library and
Information Science
San Jose State University

In Partial Fulfillment

of the Requirements for the Degree
Master of Library and Information Science

by

Richard O. Ward

August, 1992

Copyright © 1992 by
Richard O. Ward

All rights reserved

APPROVED FOR THE DIVISION OF
LIBRARY AND INFORMATION SCIENCE

Dr. Karen Ceppos
Dr. Karen Ceppos

Dr. Ruth Hafter
Dr. Ruth Hafter

Dr. Shirley Hopkinson
Dr. Shirley Hopkinson

APPROVED FOR THE UNIVERSITY

Serena M. Stanford

ABSTRACT

COPYRIGHT LAW AS IMPACTED BY CHANGES IN COMPUTER SOFTWARE TECHNOLOGIES

by Richard. O. Ward

This thesis addresses the ways in which United States copyright caselaw has been impacted by the advent of computer software. It traces the application of major copyright doctrines, especially the idea-expression dichotomy, through a selection of pivotal Federal cases.

Major policies underlying the interests being protected in the Federal courts, as well as major economic considerations (the public goods problem and network externalities), precede an examination of some current Congressional options for software protection, namely, (1) maintaining the status quo, or, (2) creating a special quasi-copyright statute applicable to software alone.

Recommendations of the 1978 National Commission on New Technological Uses of Copyrighted Works (CONTU) have marked the development of software copyright law down to the present day, and provide a framework for this paper.

ACKNOWLEDGEMENTS

To all those who have shared in my special interest for this topic, I would like to express my sincere appreciation. While their ideas have informed my own, I claim full responsibility for the content of this paper.

The helpful suggestions and encouragement of all three members of my committee have benefitted me enormously. Special thanks to Dr. Karen Ceppos, my committee chairperson, and to the other two members, Dr. Ruth Hafter and Dr. Shirley Hopkinson.

On the technical side, to Dr. C. James Schmidt, who gave so generously of his time and materials. Thanks also to Mr. David Long, a programmer at Digital Equipment Corporation, for his help in understanding the basics of how computer software is written.

TABLE OF CONTENTS

CHAPTER ONE	1
<u>INTRODUCTION</u>	1
The Statement of the Problem	2
The Hypothesis	3
The Delimitations	4
The Importance of the Study	5
Summary of the Chapters	7
CHAPTER TWO	9
<u>HISTORICAL BACKGROUND</u>	9
Historical Context	9
Historical Antecedents	12
United States Statutory History	14
The CONTU Recommendations	15
The CONTU Recommendations Reviewed	24
CHAPTER THREE	27
<u>JUDICIAL APPROACHES USED IN DECIDING SOFTWARE</u>	
<u>COPYRIGHT CASES</u>	27
The Idea-Expression Dichotomy	27
Protection of a Program's Structure, Sequence, and Organization	37
Protection of Computer Screen Displays	44
Protection of the Program's User Interface ...	46
Pending Case	53
Copyrightability of Neural Network Weights ...	59
CHAPTER FOUR	64
<u>POLICIES UNDERLYING THE JUDICIAL DECISIONS</u>	64
The Interests Being Protected	64
Economic Considerations	67
CHAPTER FIVE	75
<u>CONGRESSIONAL OPTIONS REGARDING SOFTWARE PROTECTION</u>	75
General Considerations	75
Maintaining the Status Quo	78
How Comprehensively to Act?	79
CASES CITED	85
REFERENCE LIST	87

To Pat Gallagher in appreciation of your
cheerful labors in the service of the sixth
floor community.

CHAPTER ONE

INTRODUCTION

Copyright is the Cinderella of the law. Her rich older sisters, Franchises and Patents, long crowded her into the chimney-corner. Suddenly, the fairy godmother, Invention, endowed her with mechanical and electrical devices as magical as the pumpkin coach and the mice footmen. Now she whirls through the mad mazes of a glamorous ball.

Zecharia Chafee, Jr.,
Reflections on the Law of Copyright

The Federal courts are having difficulty in applying the 1976 Copyright Act and traditional copyright law to cases involving the alleged infringement of copyrighted computer software. They have to bridge the gap that exists between the sophistication of the technology and the lack of sophistication of the law. The speed of innovation has strained the usual judicial process of drawing analogies from existing precedents so as to interpret new situations.

When the United States Congress unequivocally included computer programs in the class of created works protectible by the 1976 Copyright Act, they did not do so with the benefit of an extended history of judicial or legislative experimentation, nor even after much experience with the

subject matter. For example, the first spreadsheet program, VisiCalc, had just been introduced a year earlier to run on the Apple computer (first built in 1976), and the IBM personal computer had not yet been introduced (Shelly 1990).

This paper proposes to examine how the development of computer software has impacted United States copyright law. Creators, developers, producers, and distributors of such software are looking increasingly to copyright law rather than to other legal options to protect their works. Such law should strike a balance between the interests of such copyright owners and the need for public access to information. Currently, this balance appears threatened to the extent that Federal caselaw is overly protecting creators at the expense of innovation.

Because Federal statutory law has tended to lag behind Federal caselaw in this area, the latter will be emphasized.

THE PROBLEM AND ITS SETTING

The Statement of the Problem

This paper proposes to examine the ways in which United States Federal copyright law has been impacted by the advent of computer software technologies. An underlying problem is whether traditional copyright law is the most appropriate way to protect such technologies against possible infringement by would-be copiers.

The current trend in Federal court decisionmaking is

towards enlarging the scope of software copyright protection. Such widening of protection may be subverting important traditional functions of copyright law, namely, fostering the free flow of information and encouraging innovation. Federal courts seem to be favoring larger companies which already own important copyrights by rendering decisions holding that these same copyrights protect dimensions of the program not contemplated when the copyright was first obtained. These companies use financial resources which could be supporting research and development to support copyright litigation aimed both at major competitors and at small start-up companies attempting to enter the field. Copyright law, once used primarily as a shield against software piracy, has increasingly become a sword aimed at threats to future profits.

The Hypothesis

In 1980 the United States Congress should have passed an independent act creating a new form of copyright protection for computer software rather than merely amending the 1976 Copyright Act to incorporate this new electronic medium. An independent act could have followed the approach later used to create the 1984 Semiconductor Chip Protection Act, which established a novel category of legal protection in response to rampant chip piracy then occurring under the uncertain copyright laws.

Currently, the pressing problem with respect to computer software is not large scale piracy but large scale over-protection of software by the Federal courts. This has come about incrementally over the past two decades as protection has been extended to more and more facets of software programs. Not all courts have participated in this process, however, and this has created a schism between several Federal jurisdictions, with some relying on judicial doctrine favoring the copyright holder, and some relying upon doctrine favoring the purported infringer. The resulting confusion calls for clarifying legislation by the United States Congress in order to provide better guidance for the courts.

Congress could establish a second National Commission on New Technological Uses of Copyrighted Works (CONTU) charged with investigating the need for a new copyright act structured specifically to provide a balanced protection for both the software innovator and the public interest in free access to ideas.

The Delimitations

Aside from a brief historical summary, this study is limited to the United States. United States law provides a useful starting point for examining trends in this field given the leading role the United States has assumed in worldwide enforcement of copyright conventions.

Little attention is given to applications of trade secret or patent law in protecting computer software. Although these forms of protection can sometimes be used as alternative or simultaneous forms of protection, along with copyright law, the current turmoil in the field surrounds copyright law and its ongoing extensions into new areas of protection.

The Importance of the Study

Without computer programs the computer industry could not exist. Programs control the operation of the hardware and enable it to perform a wide variety of tasks from word processing, spreadsheets, and databases to drafting systems, teleprompters, and air traffic control. But software has a dual nature. It not only runs computers, it also conveys information. This information is packaged as software in machine readable form and is a critical commodity in our information society. One source estimated that in 1990 15.3 billion lines of software program code were written (Office of Technology Assessment 1992).

Because software is vital in expanding the demand for computer systems, the means society uses to encourage the writing of programs greatly impacts the social and economic benefits realized. Currently, copyright is the most frequently used legal device to foster software development. Yet, applying copyright law to software has created

difficulties. This is primarily because computer software is the first copyrightable work which is not intended to be 'read' when it is used. The copyrightable aspect of software is the way in which it is programmed. The creator's expression is in the programming. When one utilizes a computer, only the user interface is revealed. The expression remains concealed. Even when software is mass distributed, the programmer's expression remains hidden from the public. All copyrightable works which preceded software automatically revealed the ideas and expression in the work. Not only is the copyrightable content of software concealed from the user, but even if one goes to the Copyright Office in Washington, D.C., only a small portion of the copyrighted program's source code can be found in the public record, since the basic registration required is that of the machine-readable object code.

This limited registration, combined with an historical judicial trend towards broader copyright protection, works to leave minimal information in the public domain. To the extent that this is creating an imbalance between the use of copyright to encourage incentive, on the one hand, and the protection of the public's access to information, on the other, pressures are likely to continue to build towards legislative intervention.

The appropriateness of such intervention is likely to be impacted by future technologies as methods used to

protect software become a model for the balance of the information economy. For example, DNA, like software, both accomplishes work and contains information. It creates proteins in a living organism and contains a series of genes and other subcomponents of DNA. We may already need a system of protecting the extensive effort that goes into creating a new living organism from quick reverse engineering. "DNA companies in the 1990's will replicate the experience of software companies of the 1970s" (Davidson 1986, 1105). A key issue for software, DNA, neural networks, and other newer technologies will be how to provide for mass-marketing in a climate of uncertainty over intellectual property rights.

Summary of the Chapters

Chapter I presents the major problem which is the central focus of this paper. Also included is an examination of the the current state of copyright law.

Chapter II summarizes both European and contemporary North American antecedents to copyright law as it has come to impact current software development. After a brief introduction to English copyright law as incorporated into the United States Constitution, the focus shifts to the very important 1978 Final Report of the National Commission on New Technological Uses of Copyrighted Works (CONTU), which Congress adopted almost in its entirety as a 1980 amendment

to the 1976 Copyright Act.

Chapter III discusses several important judicial doctrines relied upon by the Federal courts in deciding copyright cases. This is done in the context of those cases which appear to have had a major impact on later cases.

Chapter IV examines the policies regarding the interests protected by the courts when deciding software copyright cases.

Chapter V proposes that a second National Commission on New Technological Uses of Copyrighted Works (CONTU) be authorized by Congress in preparation for the creation of a new copyright statute tailored to the unique nature of computer software.

CHAPTER TWO

HISTORICAL BACKGROUNDHistorical Context

The traditional 'gradualist' view of biological history has recently been challenged by a model of 'punctuated equilibrium,' in which relatively long periods of evolutionary stability are interrupted by brief episodes of radical environmental change accompanied by sudden extinction or adaptive response in the species confronted by it (Stanley 1981). An analogous process of historical development can be seen in the periodic technological threats to an author's natural right to control his or her own creation which once was protected by virtue of the inordinate labor required to copy a written work. Radical technological changes have intermittently weakened this protection. "In the entire five millenia between the invention of writing and the revolution of modern printing, successive advances in host technology reduced the cost of embedding a given intellectual good in the equivalent of a 250-page book by roughly a factor of ten" (Adelstein 1985, 223).

The invention of moveable type in the 15th century reduced the cost of this same task by a hundredfold, enabling large numbers of printers to reproduce printed material without compensating the author. The exclusionary barriers destroyed by this technological breakthrough were

eventually rebuilt by copyright statutes, products of the newly developed concept of intellectual property. Such intangible property is 'non-excludable', meaning that once it is brought forth from the mind it can generally be taken by a second user at almost no cost. Furthermore, it is, "subject to non-rivalrous consumption, in the sense that one user's use of the idea does not reduce the value of the idea to another who wishes to use it" (Carter 1990, 102). Given these characteristics of intellectual property, it would be nearly impossible for the creator to earn an appropriate return on an idea at a reasonable cost without some mechanism to ensure just compensation.

One such device is for a government to encourage innovation by paying a subsidy to the creator. Another mechanism is for the government to intervene and use law as a tool to prohibit others from taking away gains due to the creator. This is the approach of copyright law which must attempt to balance the desire of individuals and corporations to profit from their own inventiveness against the public's need to access ideas. Achieving this balance where the characteristics of the subject matter needing protection are indistinct is difficult.

"The traditional categories of intellectual property law depend on a set of clearly defined products or 'processes' - literary works, inventions, design, etc. But with the new technologies, boundaries between media are blurred and intellectual assets become increasingly abstract and intangible."
(Branscomb 1988, 41)

The same work is no longer limited to storage on paper but can now be transferred to magnetic forms such as computer tapes, discs or chips, as well as laser forms such as CD-ROMs or video discs. The information can then be transferred to large numbers of people via broadcasting, telephone lines, or fiber optics.

In the past, copyright law has been able to strike a balance between the interests of owners and the public without great difficulty because the means for copying and disseminating intellectual works were relatively unsophisticated. As information has become purer in its relative independence of any particular medium, copyright law, at least as it is evolving statutorily and in the federal courts, may no longer serve as an appropriate means for protecting the rights of creators.

"The attempt to force these new technologies into outmoded categories can create absurd and contradictory situations that threaten to undermine public confidence in the principle of intellectual property rights itself" (Branscomb 1988, 40).

Given the increasing number of infringement lawsuits, copyright owners are likely to continue to press for further revisions of copyright law to at least clarify and perhaps expand the protection afforded.

A likely candidate for such revision is the scope of copyright protection provided for computer software. "The virus now spreading throughout the industry is the ambiguity of U.S. software copyright law" (Francis 1988, 22).

Programmers are increasingly finding that in the course of their work they may be at risk of being held liable for copyright infringement (Clapes 1989). Some factors contributing to this situation include: (1) the enormous opportunities to profit from software development given the widespread use of computers; (2) increased competition among software developers; and (3) larger infringement damage awards. As to the last factor, "[W]e are seeing many more types of attorneys practicing intellectual property law...lawyers have found a gold mine in the intellectual property world" (Illinois Legal Times Roundtable 1991, 4). Opinions being generated in the resulting disputes are feeding into the legal process, (1) as software companies lobby Congress for or against legislation which would affect the scope of copyright protection for computer programs, and (2) as program authors sue one another for copyright infringement. Copyright infringement actions are producing decisions which are cumulatively redefining the parameters of copyright law and which are likely to influence any future federal statutory changes.

Historical Antecedents

The history of American copyright law began in England where the first copyright legislation known to Western civilization was instituted (Clapes 1989). By the mid-16th century the leading printers in London had formed the

Stationer's Company. In 1556 the Catholic Queen Mary granted to this company a monopoly over all printing as a means of controlling the spread of Protestantism and the Reformation. "[A]ll published works had to be entered in the register of the Stationer's Company and in the name of some particular member of that Company, who forever after had the sole right to publish that work" (Nasri 1976, 1).

The Company was charged with enforcing not only its own monopoly but also the Crown's censorship. Its wardens sought out secret presses or unlawfully printed material. Over time, however, the Company found it increasingly difficult to enforce its monopoly.

"By the end of the 17th century surreptitious presses had spread as far as Scotland and illegal secondary producers prospered, undeterred by increasingly severe penalties the Company could not reliably impose. The Stationers repeatedly petitioned Parliament for relief, and in 1710 their efforts were rewarded by the Statute of 8 Anne" (Adelstein 1985, 225).

In 1774 the House of Lords rendered an important interpretation of the Statute of Anne. In Donaldson v. Beckett (4 Burrow's Reports 248), the Lords ruled that the right to copy or publish the work vested exclusively in the author until the moment of publication, after which continued protection would rest in either author or publisher for only 28 years (a period related to the life expectancy of the times). The Company assumed that after this time period had lapsed, the right to print the work would revert back to them. Instead, the Lords ruled that

this right would henceforth pass not back to the Stationers but to the public at large, including printers who were not members of the Company. This liberation of printing from the control of the Stationers permitted a radical decentralization of the function previously carried out by the wardens. Because many publishers each held a temporary monopoly in a few works and competed in the marketplace, the responsibility for detecting illegal editions and bringing suit against infringers was spread over many motivated individuals possessing better information than had the wardens.

United States Statutory History

The Statute of Anne as interpreted in Donaldson was incorporated into the United States Constitution without debate. James Madison had written in The Federalist Papers:

"The copyright of authors has been solemnly adjudged in Great Britain to be a right of common law. The right to useful inventions seems with equal reason to belong to the inventors. The public good fully coincides in both cases with the claims of individuals" (No.43).

Pursuant to the newly-created Constitutional power, Congress passed the first copyright act in 1790, "for the encouragement of learning, by securing copies of maps, charts and books, to the authors and proprietors of such copies" (Act of May 31, 1790, 1st Cong., 2d Sess., 1 Stat. 124). Over the years this law has been amended to cover the designing of prints (1802), musical compositions (1831),

photographs and photographic negatives (1865). In 1909 Congress amended the law to consolidate these and other additions to the original copyright statute. Only three years later, a new amendment was needed to accommodate motion pictures. No further technological forms were added to the protected list until 1971 when the music industry needed protection against piracy of sound recordings.

In 1976, Congress passed another general revision of the copyright law. Lawmakers were undecided as to what amendment of the statute was needed to incorporate computer programs. To escape their dilemma, they merely recognized that such programs would have no greater protection than they had enjoyed under prior law. This evaded the issue, however, since the nature of the protection available under the old 1909 Act was highly problematic.

The CONTU Recommendations

In 1974 Congress had passed Public Law 93-573, which established the National Commission on New Technological Uses of Copyrighted Works (CONTU) to study the computer program issue. In 1975, President Ford appointed twelve CONTU commissioners according to criteria set out in the law. Two additional ex officio members namely, the Librarian of Congress and the Registrar of Copyrights, provided a total of fourteen members. This ad hoc presidential advisory commission functioned within the

Library of Congress.

Their Final Report in 1978 recommended only modest changes in the 1976 Act. These were, firstly, to add a definition of 'computer program' and, secondly, to provide that the owner of a copy of a copyrighted program did not infringe that copyright by using the program in a computer or by making an archival copy (CONTU Final Report 1978, 12). 'Computer program' is now defined in 17 U.S.C. 101 as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." Although the changes recommended by CONTU, which were adopted almost in their entirety by Congress, were modest, they sufficed to make it clear that software was protectible by copyright.

The CONTU commission was unanimous in believing that computer programs are entitled to legal protection. The commission heard testimony regarding the various forms in which programs can be prepared: Preliminarily, flow charts are often used to create a graphic representation of the data flow in the new program. Then a source code uses any of several programming languages such as COBOL or FORTRAN to map sequences in detail. Finally, an object code consisting of long strings of ones and zeros translates the source code into the machine language of the computer. All three of these forms, the commission decided, are copyrightable works of authorship, provided they require sufficient intellectual

effort to meet both a legal 'sufficient intellectual labor' test and another test which is the focus of much current debate in the copyright arena. This later test derives from the concept that no one can claim an idea and prevent others from using it in their own creations. Authors cannot be allowed to impede the progress of knowledge by monopolizing ideas that should belong to everyone. This is sometimes referred to as the 'idea-expression' test since only a creator's expression of an idea and not the idea itself is protectible by copyright.

The vice-chairman of the CONTU commission, Melville B. Nimmer, suspecting that the commission's recommendation for protection of all software might prove unduly broad, suggested that subsequent legislation might wish to, "limit copyright protection for software to those computer programs which produce works which themselves qualify for copyright protection" (CONTU Final Report 1978, 27). He suggested, for example, that a program designed for a database would be copyrightable since the resulting arrangement of items would itself be copyrightable as a compilation. By contrast, a program controlling traffic signals would not be eligible for copyright because such operation does not result in copyrightable work. This suggested limitation on program copyrightability was not accepted but does reflect an uneasiness with the sweeping protection championed by the commission, which is now under such widespread attack.

Another member, John Hersey, also wanted to limit protection. His concerns foreshadowed those of various subsequent commentators, and his contributions are cited in a number of federal judicial opinions. He noted that, "Admitting these devices [computer programs] to copyright would mark the first time copyright had ever covered a means of communication, not with the human mind and senses, but with machines..." (CONTU Final Report 1978, 28). He wanted the preliminary intellectual work such as flow charts to be protected, but not the program itself, which he saw as a mechanical device not merely giving directions for how to do work, but actually doing the work much as a machinist's cam objectifies a series of instructions to a drill. To legally call such program a copy of a literary work is also subtly dehumanizing in its blurring of the distinction, "between works that speak to the minds and senses of men and women and works that run machines - or, ultimately, the citizen's sense of the saving distinction between human beings themselves and machines themselves" (CONTU Final Report 1978, 29).

Mr. Hersey expected copyright to become yet one more legal tool to bar public access to innovative programs because much more prevalent use of trade secrecy law would continue, and use of copyright would become, "bet-hedging, reflecting efforts of major hardware manufacturers to assert any colorable claim to protection, regardless of its real

legal merits" (CONTU Final Report 1978, 34). Computer program owners would be driven into even deeper secrecy, "by encryption, physical barriers to access, contractual restraints, nondisclosure agreements, and further innovative technical tricks for locking out pirates, thieves, and competitors" (CONTU Final Report 1978, 34). Thus, public access would likely become overly restricted by adding copyright to the legal tools already available. Further, the development of protective devices would be passed onto the public in higher prices.

Mr Hersey's proposed solution was to separate programs which are copyrightable writings from uncopyrightable mechanical devices. Although this approach was not adopted, a number of his concerns have turned out to have been well founded particularly as to, (1) zealous corporate use of copyright law to guarantee maximum software profits, and, (2) use of "innovative technical tricks" by program creators to maintain secrecy, thereby inhibiting public access to program innovations (CONTU Final Report 1978, 34). These have have come to include shrink-wrap licensing and copy-protect programs.

The CONTU commission also considered and rejected the withdrawal of software copyright protection in favor of other potential forms of protection. One such form is the patent which is sometimes sought by program developers because it protects its owner not only from competition by

copiers of the work, but also from those who independently develop the same or a closely related device. Unlike a copyright, a patent protects the utilitarian aspects of a work and not merely the particular mode of expression (Menell 1987). "It gives inventors a short-term powerful monopoly in devices, processes, compositions of matter, and design which embody their ideas" (CONTU Final Report 1978, 16).

Obtaining a software patent, however, is a major undertaking. It is expensive, time consuming, and may not succeed. While patents have been obtained for certain computer programs, particularly those closely tied to carrying out industrial processes, the standards for novelty, nonobviousness, and usefulness are still uncertain (Menell 1987). A carefully prepared application must be extensively reviewed by the Patent and Trademark Office. Given the rapid pace of change of computer technology, the time required for this is rarely worth the effort. Furthermore, the patentee must make substantial public disclosures if patent protection is granted, thereby facilitating access by others to what might otherwise remain a trade secret (See 35 U.S.C. 111-112, 114, 154).

Trade secrecy is another form of computer software protection which is sometimes used, especially, "if the program is relatively expensive, and will have a comparatively modest distribution" (Bender 3.01). Trade

secrecy consists of a patchwork of common law and statutory rules which protects information that is not the object of recognized property rights (Mackaay 1990, 908). It is intended, "to protect proprietors who use a formula, pattern, device, or compilation of information in their business which gives them an opportunity to obtain an advantage over competitors who do not know or use it" (CONTU Final Report 1978, 16). The CONTU commission noted that trade secrecy is usually an inappropriate way to protect computer programs since the producer cannot maintain the secrecy of a program and hope to market it. Once the secret is disclosed, no matter what the circumstances, protection is lost.

Some computer software companies attempt to satisfy the secrecy requirement by distributing their products through limited licensing agreements rather than through sales contracts. "There is concern, however, that such agreements may be unenforceable as contracts of adhesion and that mass distribution of this type may be inconsistent with the requirements of secrecy" (Menell 1987, 1352). Such licenses are known as 'shrink-wrap', 'box-top', or 'tear-me-open' licenses. By opening the sealed software package, a customer is deemed to accept certain conditions which are visible on the package. This attempt by the software industry to avoid any transfer of ownership and to control the number of program copies the purchaser can make is an

attempt to circumvent the intent of Congress as expressed in Section 117 of the Copyright Act of 1976. This provision was designed to clarify both proprietors' and users' rights. It creates exceptions to the exclusive rights granted to the owner of a software copyright elsewhere in the statute (Section 106). These exceptions are: (1) the right to make a copy in connection with loading the software into a computer; (2) the right to make backup or archival copies; and (3) the right to adopt or modify the software (Smedinghoff 1990, 658). In making this amendment, Congress followed the recommendations of CONTU with one small change in wording. While CONTU wanted to benefit "rightful possessors" of computer software under section 117, Congress restricted the grant of rights to "owners" of copies of computer programs. While the legislative history is unclear, the wording may have been changed to bring its terminology into conformity with that of other sections of the Copyright Act (Meisner 1988, 394).

This minor change in language creates a major change in the effect of section 117. It permits software companies to restrict the level of protection available to software purchasers by characterizing such purchasers as "licensees" rather than "owners." The enforceability of the terms of such licenses, as mentioned above, is doubtful.

"When a computer user acquires packaged software from a retail outlet, the essence of the transaction is a sale. Under normal circumstances, there is no discussion about the terms of the

transfer. The salesperson prepares a sales slip, the purchaser tenders cash or a credit card, and the transaction is complete. To most consumers, receiving a sales receipt or invoice strongly implies that the title to the purchased goods has passed to them. Further, the state has exacted sales tax on the transaction; while this is typical for the purchase of goods, such a tax does not normally apply to items conveyed by license (Meisner 1988, 396).

Another form of potential software protection came into existence in 1984 when Congress enacted the Semiconductor Chip Protection Act which established a new category of legal protection for those intellectual works embodied in semiconductor computer chips (mask works) (17 U.S.C. 901-14, Supp.III 1985). The Act was mainly modelled on copyright law but did borrow certain ideas from patent law, *viz.*, "...a threshold requirement of innovation and originality not found in the traditional copyright laws." (Erstling 1989, 307). Prior to the Chip Act, an innovator was not protected by copyright law, which considered a chip design to be purely utilitarian and thus outside the scope of copyright protection. Nor was an innovator protected by patent law, since the creativity involved was insufficient to rise to the inventive level required by the patent laws. The Act was an entirely new form of intellectual property protection, "...a hybrid of traditional patent and copyright law...Although the new law incorporates the patent law's perspective on utilitarian article protection, it nonetheless adopts a traditional copyright approach to the idea-expression dichotomy" (Stokes 1985, 564). Before the

Chip Act was passed, semiconductor manufacturers faced a real threat from 'chip pirates' who engaged in blatant wholesale copying under the then uncertain copyright laws. This is now clearly prohibited.

The CONTU Recommendations Reviewed

The software industry has undergone dramatic change in the almost fourteen years since CONTU issued its recommendations to Congress. Problems which have emerged during this period of growth have highlighted some possible shortcomings of the commission's analysis and recommendations. Some of these are, "impediments to entry by small, independent firms, wasteful expenditure of vast resources to develop noninfringing, yet compatible computer systems, and the lack of clear industry standards" (Menell 1987, 1330). CONTU may have given too much credence to the idea that by providing broad protection to all forms of computer software they would, at the same time, be encouraging the optimal level of innovation in computer technology. Given the inchoate state of computer-related markets in 1978, the commission may not have appreciated the emerging division of labor in the field.

The computer industry has many interrelated stages: Basic research, operating systems, hardware products, integrated hardware systems, peripherals, applications programs, and marketing. This complex market structure

combined with the high cost of contracting among producers and consumers means that, "expansive legal protection at an early stage inhibits innovation at the other stages. This slows diffusion and adoption" (Menell 1987, 1331).

In providing sweeping protection to software creators, CONTU may not have adequately distinguished between programs written specifically for particular consumers (contract programs) and the now rapidly growing category of general purpose programs which can be used by more than one consumer (software packages). By perhaps overly protecting creators of software packages, CONTU may have inadvertently inhibited development of 'network externalities' in the computer industry. This refers to markets in which the utility that a consumer derives from a product increases with the number of other consumers of the same product. For example, the more people who own a telephone, the more each telephone owner is benefitted. The more people who use a particular operating system, the more applications programs will be developed for it. "CONTU completely overlooked the fact that operating systems serve as product standards that are capable of producing substantial network externalities" (Menell 1987, 1372). If each new operating system were less protected by copyright and, therefore, more accessible to all involved in operating system development, each system would likely be more compatible with other systems. A wider variety of applications software could run on any one

operating system, and software would be more widely available and less expensive.

CHAPTER THREE
JUDICIAL APPROACHES USED IN DECIDING
SOFTWARE COPYRIGHT CASES

The Idea-Expression Dichotomy

Products of cognitive effort can range in quality from everydry ideas, such as that of using a pencil as a letter opener, to the most extraordinary, such as the Pythagorean theorem. United States intellectual property law will not, generally, protect either category of ideas, but rather the range of ideas that fall between each extreme. Whatever the legal protection granted, it must be enforceable, and there cannot be undue interference with the use of ideas by others (MacKaay 1990, 903).

In addition, a serious constitutional question would be raised if congressional power over copyright were construed to include the power to grant monopolies over ideas (Walter 1988, 18). Article I, section 8, clause 8 of the United States Constitution gives Congress the 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." Thus, the United States Congress included a provision in the Copyright Act of 1976 stating that the scope of copyright protection does not extend to any idea. It also provided that the intellectual product must be "fixed in a tangible medium" (17 U.S.C. 101-

118, 1982). This excludes "purely evanescent reproductions such as those projected briefly on a screen, shown electronically on a television...or captured momentarily in the 'memory' of a computer" (Johnson 1990, 386). This legal concept of fixation is an outgrowth of the constitutional requirement that copyright should protect only 'writings' (Art I, sec 8, cl 8).

The outcome of copyright infringement lawsuits often depends, at least ostensibly, on whether there has been copying of the expression of an idea rather than the idea itself. "[N]o one infringes unless he descends so far into what is concrete as to invade that expression" (National Comics Publication v. Fawcett Publications, 191 F. 2d 594, 600, 2d Cir. 1951). As abstruse as is the boundary between idea and expression, the distinction "has dominated the scope of copyright protection for over a century" (Baker v. Selden, 101 U.S. 99, 102-103, 1879).

The idea-expression dichotomy is illustrated by the outcome of Synercom Technology, Inc. v. University Computing Co. (1978). Synercom had developed and marketed software that automated the task of performing structural analysis, the complex set of computations made by engineers to calculate the stresses on structures they have designed. Synercom's program, called STRAN, was introduced in 1969. It required the user to fill out one or more of thirteen forms called 'input formats', which were designed to print

out the image of an 80-column punched card, with notes as to what should be entered on the card and at what points. The format cards had titles and shaded areas delineating the proper columns in which to place certain data. The card image was given to keypunch operators who would then punch out actual cards and input them into an IBM or Univac computer. Each input format had been registered with the Copyright Office.

In 1973 EDI decided to market a program in direct competition with STRAN called SACS II. This much less expensive program was designed to allow STRAN users to conveniently switch to SACS II. To do this, SACS II came with a 'preprocessor' program that converted STRAN input formats into a form comprehensible to SACS II (Clapes 1989, 76). In developing their program, EDI had in effect translated the STRAN user manual into a computer program. The defendant had not copied the format cards but designed its program to require input of data in the same sequence as the plaintiff's program.

Synercom sued EDI for copyright infringement. Judge Higginbotham, a California Federal district court judge, posed a preliminary question, namely, whether the STRAN formats were copyrightable, ie., whether the Copyright Office had erred in accepting the input formats for registration in the first place. This depended, according to the judge, on whether EDI had "copied ideas or their

expression" (Synercom at 1012). The judge accepted EDI's argument that "because use of the idea required substantial duplication of Synercom's arrangements and sequencing, copyright protection ought not be allowed" (Synercom at 1012). Thus, EDI had appropriated ideas, not expression. The judge drew an analogy to the idea of the 'figure-H' pattern of an automobile stick shift lever to elaborate the difference between idea and expression.

The pattern (analogous to the computer 'format') may be expressed in several different ways: by a prose description in a driver's manual, through a diagram, photograph, or driver training film, or otherwise. Each of these expressions may presumably be protected through copyright. But the copyright protects copying of the particular expressions of the pattern, and does not prohibit another manufacturer from marketing a car using the same pattern (Synercom at 1013).

EDI's argument was and still is referred to as the 'merger' doctrine which holds that "a work is uncopyrightable if its idea and expression merge, or are so closely related that one cannot be used without the other" (Leaffer 1989, 69).

If an idea is 'merged' into its expression, the expression cannot be protected under the copyright laws since no expression separate from the idea exists.

In explaining the merger doctrine the courts sometimes cite Herbert Rosenthal Jewelry Corp. v. Kalpakian (446 F. 2d 738, 741-2, 9th Cir., 1971) which held that,

the idea of a jewel-encrusted bee pin could not be distinguished from the expression of that idea. As long as more than one manufacturer produced jewel-encrusted bee pins...it was inevitable that there would be a substantial similarity between the two forms of expression. In other words, the number of ways in

which one can design a jewel-encrusted bee pin is strictly limited. Under such circumstances, to give one manufacturer a copyright on its bee pin would be to give that manufacturer a monopoly on the jewel-encrusted bee pin market because no other manufacturer could possibly conceive a substantially different jewel-encrusted bee pin.

Thus, if an idea can only be efficiently expressed in a few ways or in only one way, the idea and its expression are said to merge, making the expression necessary to the idea.

The theory behind the merger doctrine is that if an idea were copyrightable, a monopoly could be created that would occupy an entire area of thought, preventing anyone else from expressing such ideas. "To avoid this type of protection, copyright does not extend to an idea or to an expression whose idea has merged with that expression" (Johnson 1991, 951).

Synercom had argued in vain that the sequencing and ordering of data on their program input formats was copyrightable expression. The judge responded, "Synercom's argument is double-edged. If sequencing and ordering is expression, what separable idea is expressed?" (Synercom at 1013) Apparently finding no separable idea if the input sequence was construed as expression, the judge implied that the sequencing must be an idea.

Several later courts, most notably Whelan Associates v. Jaslow Dental Laboratory (1986), have rejected this reasoning, and instead found program sequencing to be copyrightable expression rather than unprotectable idea.

Recently, the judge in Lotus Development Corp. v. Paperback Software International (1990) also refused to follow Synercom, noting its failure to cite any portion of the CONTU report, published a month earlier, and, in addition, the subsequent Congressional passage of the 1980 amendments to the Copyright Act.

The search for the idea-expression boundary in copyright cases has proven to be more difficult in cases such as Synercom involving computer programs than in cases involving more familiar subject matter such as literary works. Computer programs are not read by the general public, but are only used by them. Most people are not trained in programming languages. This leaves it to judges and juries to categorize seemingly esoteric programs interpreted by expert witnesses using unfamiliar terminology. Since most people are more familiar with human languages than with programming languages, analogies between the two are sometimes made to better conceptualize what has come to be called the 'idea-expression dichotomy'.

One commentator compares the source code of a program (eg., BASIC or FORTRAN) to the surface structure of a human language. The programmer uses this structure to express meaning within the computer's linguistic universe (Drysdale 1987). The source code uses English words and symbols, and is relatively easy to learn and understand (e.g., "GO TO 30" tells the computer to skip intervening steps and go to the

step at line 30). Programs written in source code can be translated by a "compiler" program into object code for use by the computer. The object code (also called 'machine language') uses long strings of ones and zeros to make the source code 'readable' by the computer. These ones and zeros are in the form of a set of electrical impulses that, more or less, correspond to the source code. The object code can be conceptualized as the deep structure of the source code. It is the object code which carries the core meaning of the commands to the computer. It is the object code which approaches the idea of a program as opposed to its expression in source code.

The copyrightability of source code has been accepted since the advent of electronic computer programs. However, protection of object code has generated controversy, especially since the CONTU Report was issued in 1978 (Samuelson 1984). Whether such machine language, the program's 'idea', should be protectible by copyright was a source of division among the CONTU commissioners and has been important in the precedent-setting Apple Computer cases.

The disclosure problem is a relatively recent one. Until electronic computer programs existed, copyrighted works that were sold to the public communicated the ideas they contained. "Computer programs in machine-readable form are the first type of copyrightable work to have a major

commercial value without disclosure" (Samuelson 1984, 710). Source code, unlike object code, does provide some disclosure since it can be read by someone who understands the language or symbols the programmer used to describe the program. To maintain competitive advantage, protect proprietary rights, and avoid redundancy, only the machine-readable object code and not the source code is generally present in software sold to the public (Samuelson 1984, 634 n.67).

The Copyright Office requires that only a small portion of the full text of the source code be deposited with the Office so that it can identify the particular program with greater certainty than if the object code alone were filed (See 37 C.F.R. @202.20(c) (vii)). With enactment of the 1976 Copyright Act, deposit of the full text of the source code "as the best representation of the authorship in the program..." was no longer required ("Notice of Inquiry [Regarding] Deposit of Computer Programs and Other Works Containing Trade Secrets," 48 Fed. Reg. 22, 951, 1983).

Arguably, providing for copyright registration of only the object code of a program discloses almost nothing in return for the protection of the law. The public may not be receiving enough in exchange to warrant the seventy-five year monopoly that corporate owners of published copyrighted programs receive. A few courts have been so troubled by the fact that a software producer was permitted to copyright a

program without being required to first make the complete source code publicly available as a quid pro quo for the United States government's copyright protection, that they have denied relief on copyright claims (Data Cash Sys., Inc. v. JS&A Group, Inc., 480 F. Supp. 1063, 1071, N.D. Ill. 1979, 628 F. 2d 1038, 7TH cir. 1980; Apple Computer, Inc. v. Franklin Computer Corp., 545 F. Supp. 812, 323-25, E.D. Pa. 1982, rev'd, 714 F.2d 1240, 3d Cir. 1983).

The district court in Apple Computer, Inc. v. Franklin Corp. did not find that object code disclosed enough information to human beings to be regarded as a copyrightable 'writing' under the Constitution.

Enlarged as the concept of "writings" now is, it retains its original meaning: to express and communicate...It is not clear in this case that an operating system in binary code or one represented either in a ROM...are "explanations"...If the concept of "language" means anything, it means an ability to create human interaction (545 F. Supp. 824, E.D. Pa. 1982).

This case is important since, when reversed on appeal, strong precedent was created for protecting machine-readable object code as well as human-readable source code. At the initial trial court level, Franklin Computer Corporation ('Franklin') admitted it had copied fourteen software programs produced by Apple Computer, Inc. ('Apple'). The sole issue was whether the software was protected by copyright. The court viewed the copied operating systems as "an essential part of the machine that makes it work" (Apple 1982, 812, 815). This was by way of

saying Apple's software could only have been protected by patent and not by copyright. The court refused to grant Apple's request for a preliminary injunction to stop Franklin from selling its Ace 100, a personal computer which contained the fourteen operating system programs expressed in object code and either stored on Ace ROM's (Read Only Memory chips) or on floppy disks sold with the Ace. These programs made the Ace compatible with the Apple II computer so that the abundant applications software created for the Apple II could also be used on the Ace.

After being denied an injunction against Franklin, Apple appealed. On appeal the court ruled for Apple. It rejected the lower court's distinction between operating system programs and application programs:

Both types of programs instruct the computer to do something. Therefore, it should make no difference... whether these instructions tell the computer to help prepare an income tax return (the task of an application program) or to translate a high level language from source code into its binary language object code form (the task of an operating system program such as "Applesoft" [one of the fourteen copied programs]) (714 F. 2d 1240, 3d cir. 1983).

The court went on to point out that the statutory definition of a computer program in the Copyright Act makes no distinction between application and operating programs.

Just prior to publication of this appellate opinion, the chairman and chief executive of Franklin, R. Barry Borden, expressed what he saw as the larger issue in the case, namely, "whether the copyright laws can be stretched

so far as to afford protection in areas for which they were never intended." (Borden 1983, 4) He saw Apple as seeking an unwarranted monopoly in high technology: "If Apple's view prevails, rather than Franklin's, prices for computers will be higher, fewer computers will be sold, and the software industry will have a smaller market. Unhappily, the consumer will pay more for technology" (Borden 1983, 4). Whatever the truth of the economic consequences of Apple's subsequent victory in court, concerns regarding undue 'stretching' of copyright law are present in many computer software cases.

Protection of a Program's Structure, Sequence, and Organization

Whelan Associates v. Jaslow Dental Laboratory (1986) stretched copyright law further than had earlier cases. The key issue was whether only literal copying of the computer code was proscribed or whether, beyond this, the overall structure, sequencing, and organization of the program was protectible by copyright. In Whelan neither the source code nor the object code had been copied. Indeed, each of the two contending parties "had created the two programs in different programming languages on different computers..." (Fetterman 1986, 13). At issue here was which elements of a copyrighted program constituted protectible expression.

The case grew out of a series of disputes over who

owned copyright rights in a jointly-created software package called Dentalab. This package was sold to dental laboratories to automate their ordering, inventorying and accounting. The program had originally been written by Elaine Whelan in Event Driven Language (EDL) for an IBM/Series 1 computer. She had prepared for this task by studying the accounting practices of Jaslow Dental Laboratory, Inc. (Jaslow). Initially Whelan had intended to market the program but she later agreed that Jaslow would market and sell the program for a commission. After several disagreements, the arrangement was terminated, but not before Mr. Rand Jaslow obtained a copy of the Dentalab System source code without Whelan's consent (Whelan at 1314). Using this copy of the source code and the assistance of an expert programmer, Jaslow developed a dental laboratory program in the BASIC programming language for an IBM PC computer (Whelan at 1320). The two programs were substantially similar, not only in function, but also in the detailed design, logic, structure, and flow of five important subroutines. They were also substantially similar with regard to data structures and screen displays (Whelan at 1228).

Both the Federal district and appellate courts held that Jaslow had infringed upon Whelan's copyright. The upper court's decision, which has become a milestone in copyright law, held that anyone is free to copy whatever is

needed to achieve the same purpose as an original program, here the efficient organization of a dental laboratory. But if one goes beyond this purpose or idea and copies the 'structure, sequencing, and organization' of the other program, one may be infringing on another's copyright, just as copying the structure, sequencing, and organization of a copyrighted novel may be an infringer, "One can violate the copyright of a play or book by copying its plot or plot devices" (Whelan at 1234 citing Twentieth Century-Fox Film Corp. v. MCA, Inc., 715 F. 2d 1327, 1329, 9th Cir. 1983). The court quoted Justice Learned Hand in Nicholas v. Universal Pictures Corporation (45 F.2d 119, 121, 2d Cir. 1930), "[Copyright] cannot be limited literally to the text, else a plagiarist would escape by immaterial variations".

By protecting the effort and expense that had gone into Elaine Whelan's source code, the court was favoring the protection of incentive over free dissemination of information. "In this regard, we must remember that the purpose of the copyright law is to create the most efficient and productive balance between protection (incentive) and dissemination of information, to promote learning, culture and development" (Whelan at 1235). The court accomplished this by extending protection of computer programs "beyond the programs' literal code to their structure, sequence, and organization" (Whelan at 1248).

The court also relied upon the more traditional idea-

expression test to reach the desired result: "The conclusion is thus inescapable that the detailed structure of the Dentalab program is part of the expression, not the idea, of that program" (Whelan at 1239). While acknowledging that its interpretation of the idea-expression test "may put us at odds with Judge Patrick Higginbotham's scholarly opinion in Synercom Technology, Inc. v. University Computing Co." (Whelan at 1240), the court answered that,

"[T]he idea is the efficient organization of a dental laboratory...Because there are a variety of program structures through which that idea can be expressed, the structure is not a necessary incident to that idea" (Whelan at 1240).

In response to defendant Jaslow's argument that it is the structure of the computer program itself which constitutes the idea, and not the purpose of the program, the court responded that by comparing similarities between the two programs' overall structures, sequences, and organizations, it was not improperly extending Whelan's copyright to the underlying ideas embodied in Dentalab. This was because, according to the court, the structure, organization, and sequence was expression (Hoffman and Grossman 1990).

The application of the idea-expression dichotomy test by the Whelan court has been criticized. By defining the efficient organization of a dental laboratory as the idea, "everything in a program at a lower level of abstraction amounts to protected expression" (Cruz 1990, 247).

Arguably, the Whelan approach ignores the way in which computer programmers write programs. They start with analysis on an abstract level and proceed to more concrete problem-solving steps that put many ideas, not one single idea, to work. By defining idea at the level of the overall function of the program and by ignoring the presence of many ideas in a program, the court may have gone too far in denying other programmers access to ideas used in the original program. "The Whelan case...stands as a major example of the improper application of the idea-expression dichotomy, because it fails to apply the limiting principle of the dichotomy that ideas are not protected" (Cruz 1990, 247).

While Whelan has established strong precedent for protecting the overall structure, sequence, and organization of software, authority for a narrower scope of copyright protection exists in Q-Co Industries, Inc. v. Hoffman (Q-Co) (1985), which was decided almost simultaneously with Whelan by a different Federal court. In Q-Co the program's structure, sequence, and organization was not protected. Both the program owned by Q-Co and the purportedly infringing program were designed to permit the use of personal computers as prompters for television and theater presentations. Such prompters scroll large characters on a television screen in front of a speaker. This allows the speaker to appear not to be reading while actually reading a

prepared text on the screen.

The defendant, Hoffman, while an employee of Q-Co, had primary responsibility for developing the prompter software for an Atari 800-XL computer. This computer has a powerful graphics chip built into its hardware which permits the generation of large characters and the scrolling down through them by a simple sequence of program commands. After completing design of VPS-500 software for the Atari computer, Hoffman and a friend formed Computer Prompting Corporation (CPC) in order to create similar software for IBM computers, which had a larger memory for holding more text. Because such computers lacked a powerful graphic chip, like that in the Atari 800-XL, much additional programming work was required. When CPC presented their partially completed CPC-1000 software for the IBM computer at a 1985 National Association of Broadcasters show in Las Vegas, a 'shocked' Q-Co sued both Hoffman and CPC for infringement of the Atari program Hoffman had created for them while their employee.

The judge examined the nature of the similarities of the Atari VPS-500 and the IBM CPC-1000 software. It concluded that, "the evidence established similarity of 'ideas' rather than expression, and that, while the defendants employed 'the structure, function and concept of the VPS-500 program, CPC-1000 was not a 'paraphrase' or 'copy' in the infringement sense" (Walter 1988, 125). The

court held, therefore, that CPC had not infringed Q-Co's copyright, because CPC had merely copied the unprotectable idea of the VPS program.

The decision emphasized that literal copying had not been possible because the two programs were written in different programming languages. "...[I]ndeed direct copying was impossible since the Atari programming was in Basic and the IBM in Pascal" (Q-Co at 615). The court applied the idea-expression dichotomy test and reasoned that although some of the program subroutines (modules) corresponded closely to one another, they would be inherent in any prompting program and therefore constituted unprotectable ideas (Q-Co at 616). "Their [the subroutines] order and organization can be more closely analogized to the concept of wheels for the car rather than the intricacies of a particular suspension system" (Q-Co at 616).

Since this was a district (trial) court decision, and not rendered by a Federal appellate judge, it is not binding on any other court, and, therefore, may not be strong legal precedent for future cases.

The decision has been criticized for purportedly misapplying the idea-expression dichotomy test, and for arbitrarily refusing to protect nonliteral copying:

The Q-Co court dismissed the importance of similarities in structure by stating that differences in hardware made structural copying impossible. The Q-Co court also failed to recognize that a programmer can appropriate protectible expression in a program even though a programmer modifies or adapts the program

to operate on a different computer. Whelan... established that such nonliteral copying of a program constitutes copyright infringement" (Fetterman 1986, 25).

Subsequent courts have tended to follow Whelan's more protectionist approach to copyright rather than Q-Co, especially in extending protection to previously unprotected screen displays.

Protection of Computer Screen Displays

The first indication that the Federal courts might extend copyright protection to computer screen displays came in the Whelan case. The defendant in that case had objected on appeal to the lower court's comparison of the computer screen displays produced by the two programs at issue in determining the similarity of the underlying programs. While the court agreed that the screen output would not be direct evidence of copying, it refused to disregard screen display similarities entirely in weighing the evidence (Whelan at 1234). "The screen outputs must bear some relation to the underlying programs, and therefore they have some probative value" (Whelan at 1244, emphasis in the original).

In this portion of the opinion, the court went on to open the floodgates for subsequent 'look-and-feel' lawsuits by its assertion that screen outputs could constitute indirect evidence of infringement of the copyright on the underlying program. The court dismissed concern that

"because the screen outputs are vivid and easily understood (at least as compared with the obscure details of computer programs), they might have disproportionate influence on the trier of fact [ie., judge or jury]" (Whelan at 1245). The court acknowledged that programs relating to screen outputs are often a small part of the full program and may say very little about the underlying program. But, it went on, "[s]creen outputs are not so enticing that a trier of fact could not evaluate them rationally and with a cool head. The party against whom the screen outputs are introduced can easily explain their limited probative value" (Whelan at 1245).

The court in Broderbund Software, Inc. v. Unison World, Inc. relied upon Whelan rather than Synercom in its approach to distinguishing between idea and expression. It believed that the court in Whelan had adequately answered Judge Higginbotham's question in Synercom implying the need for some 'separable idea,' by stating that the separable idea of the dental lab program was the efficient organization of a dental laboratory, which could be expressed in various ways through a variety of computer program structures.

Likewise, in the present case, the separable idea of 'Print Shop' is the creation of greeting cards, banners, posters and signs that contain infinitely variable combinations of text, graphics, and borders. A rival software publisher is completely free to market a program with the same underlying idea, but it must express the idea through a substantially different structure (Broderbund at 1132).

The case involved two competing menu-driven programs

that enabled their users to create customized greeting cards, signs, banners, and posters. The plaintiff, Broderbund Software, successfully argued that the overall appearance, structure, and sequence of the visual displays in defendant's 'Printmaster' infringed upon their copyright on 'Print Shop' (Broderbund at 1127). The court observed:

Put simply, 'Printmaster' looks like a copy of 'Print Shop,' with a few embellishments scattered about in no particular order. The 'total concept and feel' of these programs...is virtually identical...The ordinary observer could hardly avoid being struck by the eerie resemblance between the screens of the two programs" (Broderbund at 1136, 1137).

Like Q-Co, this was a district court decision, and not rendered by a Federal appellate judge. It is, therefore, not binding on any other court.

Protection of the Program's User Interface

In a more recent district court decision, Lotus Development Corporation v. Paperback Software (1990), Lotus sued Paperback alleging that the latter's VP-PLANNER spreadsheet program infringed the copyright on the LOTUS 1-2-3 spreadsheet. The court found that Paperback's copying of copyrightable portions of LOTUS 1-2-3 to have been "overwhelming and pervasive" (Lotus at 68). Defendants testified that they had decided to make VP-PLANNER 'compatible' with LOTUS 1-2-3 and that, "[t]he only way to accomplish this result was to ensure that the arrangement and names of commands and menus in VP-PLANNER conformed to

that of LOTUS 1-2-3" (Lotus at 68).

Behind the dispute was the fact that corporate purchasers of such spreadsheets spend hundreds of millions of dollars every year training employees to use existing software. Such companies are unlikely to purchase improved programs if this requires major retraining. "Thus, imitation of user interface is not a form of flattery, but of survival" (Walter 1988, 149).

Although Lotus is often referred to as a 'look and feel' case (ie., that infringement should be found where one program has the same gestalt or overall 'look and feel' as the other), the court chose to focus instead on the 'user interface' as a sub-set of 'look and feel'. By bypassing the 'look and feel' concept and adopting 'user interface' instead, the court was able to avoid having to respond to the various ways in which prior courts had used this term and to present its own approach.

The 'user interface' of a software program is the means through which the user communicates with the program. Historically, user interface technology has evolved from the paper punch card to the modern user interface in which a user "types commands at a video display terminal or screen, presses specially-designated function keys, or uses a mechanical pointer called a mouse to select a command on the screen" (Lewis 1990, 693).

The Lotus court adopted the plaintiff's contention that

'user interface' should include, "such elements as 'the menus (and their structure and organization), the long prompts, the screens on which they appear, the function key assignments, [and] the macro commands and language" (Lotus at 63).

In Lotus, Judge Keeton posed a three-part test for determining infringement. The first part of the test requires a definition of the idea involved along a sliding scale that runs from generalized ideas to progressively more particularized expression. At the most general level, the idea was held in Lotus to be a computer program for an electronic spreadsheet. This definition was expressly based on the broad protection given to nonliteral elements of computer programs found in both the Whelan and Broderbund cases, where the idea was defined very narrowly as the purpose of the program.

Thus, even though programs like VisiCalc, 1-2-3, Multiplan, SuperCalc4, and Excel are very different in their structure, appearance and method of operation, each is, at the most basic level, just a different way of expressing the same idea: the electronic spreadsheet (Lotus at 64).

Since an electronic spreadsheet was defined as idea and, therefore, not expression, it was uncopyrightable.

The second part of the test examined whether the alleged expression is only one of a limited number of ways of expressing the idea, or whether there are other ways to express the same idea. In the former case, merger would occur and protection would be precluded. In the latter

case, the creative expression could receive copyright protection. The marginal portion of the spreadsheet screen display resembling the letter "L" (rotated ninety degrees to the right) was found to be essential enough to such display to be a part of the non-copyrightable idea. The same was true of the use of a slash ('/') key to elicit the menu command system. This was held to be one of the few practical options available, since the number and letter keys were needed as numerical values and for labeling. Also, the key had to be easily accessible because it was used frequently. Thus, "this expression [using the slash key] merges with the idea of having a readily available method of invoking the menu command system" (Lotus at 66). In the same way, the hyphen ('-'), plus ('+'), asterisk ('*'), slash ('/'), and 'Enter' keys were not copyrightable because they were expressions which existed in all electronic spreadsheets.

What was not necessary to the idea of any spreadsheet, and thus protectible expression, was "the precise 'structure, sequence, and organization,' Whelan, 797 F. 2d at 1248, of the menu command system" (Lotus at 67). This was true even though many specific command symbols were the same in both VP-PLANNER and LOTUS 1-2-3.

The fact that some of these specific command terms are quite obvious or merge with the idea of such a particular command term does not preclude copyrightability for the command structure taken as a whole (Lotus at 67).

Since the court held that the menu command structure was copyrightable expression, it applied the third part of the test, which was to determine whether this expression was a substantial part of the work.

If...the expression of an idea has elements that go beyond all functional elements of the idea itself, and beyond the obvious, and if there are numerous other ways of expressing the non-copyrightable idea, then those elements of expression, if original, are copyrightable" (Lotus at 59).

Applying this third and last part of the test, the court found that the overall menu command structure was copyrightable expression. This structure included the order of commands on the menu lines, the choice of letters or words to correspond with each command, the screen presentation of the menu lines, the type of menu used, and the long prompts. All of these could be expressed in many ways and the defendant, Paperback, had chosen to maximize compatibility by developing a 'substantially similar' structure. Indeed,

the VP-Planner manual could truthfully declare:
"VP-Planner is designed to work like Lotus 1-2-3,
keystroke for keystroke...VP-Planner's worksheet
is a feature-for-feature workalike for 1-2-3...
It has the same command tree...Everything 1-2-3
does, VP-Planner does" (Lotus at 69).

The court held that the great effort defendants had put into copying Lotus 1-2-3 was itself evidence of the substantiality of copying.

The court refused Paperback's invitation to rule in their favor as a means of encouraging software standardi-

zation. Paperback argued that the menu structure of Lotus 1-2-3 had set a de facto industry standard for all electronic spreadsheets. Thus, they maintained, there was no other choice but to copy such structure. The court held that such reasoning ignored the commercial success of EXCEL, an innovative spreadsheet program that is not compatible with LOTUS 1-2-3 in its menu structure. It also held that one object of copyright law is to protect expression in order to encourage innovation. "By arguing that 1-2-3 was so innovative that it occupied the field and set a de facto industry standard, and that, therefore, defendants were free to copy plaintiff's expression, defendants have flipped copyright on its head" (Lotus at 79). Furthermore, it concluded that there is no statutory authority indicating that standardization is necessarily in the public's best interest. The court used the shortcomings of the QWERTY typewriter keyboard as an example which calls into question the desirability of early standardization (Lotus at 79).

The court thus rejected Paperback's argument that the policy of encouraging standardization mandates a finding that a user interface is not copyrightable. In the process of examining this proposed policy, the court saw itself as almost assuming an inappropriate legislative role:

Policy arguments advanced without focusing on whether they are inconsistent with the mandates of legislation...are fundamentally flawed because they do not take account of limitations on the role of courts, to which counsel as well as courts are obliged to be sensitive (Lotus at 71).

Paperback perhaps could have argued that standardization is not inconsistent with the primary goals of copyright law: "encouraging innovation through a system of private rewards and serving the public interest in widespread access to creative works" (Lewis 1991, 691). In perhaps overly emphasizing the former goal, innovation, the court may have overlooked the clear public interest in widespread access to standardized user interfaces.

Such standardization provides familiar methods for using software programs, lowers training costs to businesses, and facilitates user access to information in libraries and other information centers. Standardization may also provide users with a wider selection of products, since the more users there are who use a particular standard, the more likely it is that software developers will create products for that standard.

Since a predominant goal of the 1976 Copyright Act is to serve the public interest in accessing creative works, Lotus perhaps should not have been allowed a monopoly on their widely used arrangement of commands and menus. "Based on sales statistics and market share, 1-2-3 appears to have risen beyond mere bestseller status and turned into a standard" (Lewis 1991, 660). The Lotus monopoly on this inchoate standard could impede the recent trend toward standardization of user interfaces in application programs in general (Giannetti 1990).

The only example the Lotus decision provides of how standardization does not serve the public interest is that of the QWERTY typewriter keyboard arrangement. It is true that the original purpose of this arrangement no longer exists, namely, to slow typists down so that the mechanical hammers of early typewriters would not stick together. Nevertheless, this standard arrangement has served the public interest by providing a familiar, predictable layout on all English-language keyboards. Following the Lotus anti-standardization reasoning to an extreme,

each typewriter and keyboard manufacturer would be encouraged to develop keyboard layouts which were technologically 'innovative' but most likely incompatible ... Merely because a standard is not the most technologically elegant does not automatically mean that the standard cannot serve the public interest (Lewis 1991, 713).

The court could have factored standardization into its idea-expression analysis by defining the idea of the computer program not simply as a spreadsheet, but rather "as a generally saleable, marketable spreadsheet where a key ingredient of marketability is adherence to the prevailing standardized user interface" (Lewis 1991, 713).

Pending Case

On March 17, 1988, Apple Computer, Inc. (Apple) filed suit against Microsoft Corporation (Microsoft) and Hewlett-Packard Company (HP) for copyright infringement. Apple originally claimed that Microsoft's Windows 2.03 program,

designed to make IBM-compatible computers easier to use, and HP's New Wave program, copied the "look and feel" of the screen displays in its user interface (Zachman 1992, 74). These screen displays include features such as 'file folder,' 'drag and drop,' 'zooming,' and 'overlapping windows,' all of which direct the user through different tasks.

Apple later added the 3.0 version of Microsoft's Windows to its suit, claiming that both the Windows 2.03 and Windows 3.0 programs were not covered by a 1985 license which allowed Microsoft to use some of Apple's copyrighted features in Microsoft's 1.0 program. In an April 15, 1992, ruling, District Court Judge Vaughn Walker disagreed. He held that of the 59 visual elements in dispute, 26 were either (1) new innovations in Windows 3.0 and were not, therefore, infringing, or (2) were covered by the 1985 license, or (3) were not protectible because they were not original to Apple, or (4) were unprotectable because their expression had merged into their ideas. An additional ten visual elements were held to be in the previous version of Windows 2.03 and covered by the license agreement.

At this time, the judge has not ruled on twenty-three small visual elements in the case. These are referred to as the 'Paragraph X' or 'Paragraph Y' elements (KQED, National Public Radio, "Forum", Apr. 16, 1992). One of these, for example, reads: "Icons are generally arranged in rows or

columns in one of the first displays that the user sees and which the user uses to select and initiate work activity" (KQED, Apr. 16, 1992). As to these Paragraph X and Paragraph Y elements, "Microsoft and HP will argue that these are all very functional [ie., protectible only by a patent] and are not entitled to copyright protection (KQED, Apr. 16, 1992).

The April 15, 1992, decision surprised many observers. During the first few months of 1992, many expected Apple to emerge victorious. One commentator wrote in mid-February: "There now seems to be considerable legal sentiment that Apple stands a very good chance to win the suit against Microsoft" (Zachman 1992, 74).

According to a damage-estimate figure released by Microsoft in early February, 1992, "Apple is seeking \$3 billion as the result of reduced sales and depressed selling prices for its Macintosh computers" (Quick 1992, 16). Apple, however, stated that they would not make a claim for damages until the case went to a jury. Thus far, the judge has ruled against Apple as a matter of law and has not found issues of fact to send to a jury.

Prior to the April 15, 1992 ruling, there were fears that Apple might try to force Microsoft to stop selling Windows and to recall any unsold software. Had this occurred, it could have disrupted the computer-based activities of many users. Microsoft has thus far sold 10

million copies of the Windows software, and it has become integrated into mission-critical applications at major corporations (Daly 1992). The April 15 decision has greatly lessened the significance of Apple's lawsuit. In the aftermath of the ruling favoring the defendants, Microsoft stock rose \$11.87 1/2 in one day (News Services 1992, 5).

Several attorneys believe that Apple was unsuccessful because it,

refused to sufficiently detail which items were copyrightable and which were not, and which Microsoft and HP were accused of infringing. Without that step, Walker [the judge] said, it could not claim an overall infringement on the Macintosh's 'look-and-feel' (Kostel 1992, 1).

The judge who preceded Walker, William Schwarzer, who subsequently withdrew from the case, "had asked Apple in previous orders for lists of similarities between the programs, indicating that an element-by-element analysis was necessary" (Orenstein April 17, 1992, 1). Apple is believed to have not complied in full due to competitive considerations, and the company pressed to have the case sent to a jury to determine if the look-and-feel, gestalt, or arrangement of the defendants' programs was infringing.

A week prior to the April 15 ruling, a Federal Appeals court had upheld the judgment in another case on the grounds that a comparison of individual elements of the programs at issue was a correct test of copyrightability (Orenstein April 17, 1992, 1). Apple has cited this decision in support of its position. That ruling, Brown Bag Software v.

Symantec Corp. (9th Cir., No. 89-16239, April 7, 1992), was favorably cited by Walker in the April 15 decision. There is a footnote in Brown Bag which could assist Apple on appeal. It reads:

Although copyright protection is not afforded to certain elements of a work, such limitations must not obscure the general proposition that copyright may inhere...in the selection and arrangement of unprotected components (Brown Bag, note 4).

Thus, Apple could argue, Judge Walker may have overly emphasized the individual components of the visual user interface as opposed to the overall look-and-feel of the programs at issue.

Subsequent to the April 15 ruling, Apple filed a motion for reconsideration claiming that Microsoft and HP had made misstatements and apologized to the court, "for not having previously organized the evidence on an item-by-item basis and any other deficiencies perceived by the court in our prior submissions" (Orenstein, May 12, 1992, 3). Despite this new willingness to engage in what is being called 'analytic dissection,' Apple still continued to maintain that the overall arrangement or 'look and feel' of the MacIntosh user interface is protectible, whether or not the elements would be in isolation.

Judge Walker stated that Apple's failure to provide the detailed analysis earlier was, "an amazing tactical decision" (Orenstein, May 13, 1992, 1), but reluctantly agreed to reconsider his April 15 ruling. He said that he

was "inclined to reject the motion for reconsideration out of hand" (Kehoe 1992, 27), but, recognizing the importance of the case to the entire computer industry, and perhaps not wanting to prejudice the client, Apple, for the actions of its counsel, he agreed to reconsider his earlier ruling at a future date.

Placing each party's situation in a broader context, Ronald Laurie, an intellectual property attorney who worked for HP early in the Apple case, said,

It is as if we are operating in a sea of free competition where there are individual islands of copyright protection. Unless an innovator can get himself on one of these islands, he is in the sea and subject to unrestricted competition" (KQED, National Public Radio, "Forum", Apr. 16, 1992).

Whether Apple has succeeded in enlarging its island should become clearer as the case progresses.

Henry C. Bunsow, an attorney who represented Xerox in an earlier unsuccessful copyright suit by Xerox against Apple, believes that a final ruling against Apple will open the market for others who want to adopt a graphics-based user interface.

If independently developed, other graphics-based interfaces will be allowed, and indeed, will become a standard way of relating to one's computer. ...This ruling will clearly encourage other developers. I think probably every computer system five years from now will have some type of graphic user interface (Kostal 1992, 1).

Given industry pressure to make graphical user interfaces the standard way in which people interact with computers, the final decision in Apple may establish a precedent which

will greatly influence how closely companies competing with Apple can copy Apple's visual interface. This may, in turn, impact the degree of standardization available to users.

Copyrightability of Neural Network Weights

During the last six years, increasing work has been done on a kind of computing architecture called 'neural networks'. This unique technology had its origins in the early 1940's, when scientists built working models to test simple electronic versions of biological brains. In the 1950's, scientists at Harvard University developed an 'electronic rat' capable of learning to work its way through a maze. The digital computer revolution overshadowed such neural network research until recently when frustration with the limitations inherent in conventional digital computers in such areas as robotics and pattern recognition has revived interest in neural networks (Wensky 1991).

A neural network consists of a large number of interconnected processing elements arranged in layers. As electronic signals pass between the layers of the network, they are modified by 'weights,' which are voltage levels under which the 'neuron' cannot fire. When the integrated input exceeds the weight, the neuron will fire, thus passing its electrical charge to another neuron at a different level of processing elements. The weights are not assigned by a human programmer. Instead,

[n]eural networks are 'trained'. The network is presented with input values for which the desired output is known. The network then adjusts the weights until this output is achieved. This process is repeated for a large number of input and output examples, called a training set. Given enough examples, the desired behavior of the network can be achieved for a wide range of inputs (Office of Technology Assessment 1992, 152).

No human tells the network the rules relating the input data to the output data. Rather, the network learns them itself as a result of the training it receives.

For real-world networks, several hundreds or thousands of facts are required, and it may take dozens of hours of computer time to feed in all the facts hundreds or thousands of times before the network adjusts its weights to get the correct answer for all the facts (Johnson-Laird 1990, 11).

The specific configuration of these adjustments cannot be predicted, yet, by analogy, these weights can be said to represent the 'program' of a neural 'computer'.

Recently, Verifone, Inc. unveiled its new 'Onyx' system that uses a neural chip to read the string of account numbers printed along the bottom of checks. The device is expected to be used in stores and banks to instantly verify that a patron has sufficient funds to cover the check.

Inside each Verifone unit is a neural chip containing an electronic grid. When a check is slid through the Onyx unit, each number on the check is briefly projected onto the grid by special lights, creating different voltage levels at each grid point. The chip is able to recognize each number because it has been 'taught' in advance which grid patterns correspond to which numbers (Gomes 1992, 1E).

Other uses for neural networks being researched include inspecting products as they come off an assembly line,

identifying currency denominations, and sorting mail.

Neural weights are a new way to embody knowledge and information, a new form of intellectual property to which patent and copyright laws will have to adapt. While about 200 neural network patents have been issued in the United States since 1963, the first copyright registration for a set of neural weights was issued in October, 1990 (Office of Technology Assessment 1992).

Statutory law requires that a literary product be 'an original work of authorship' (17 U.S.C. sec. 101) in order to be protected. Arguably, a machine, and not a human being, actually authors the weights in a neural network, since the human operator merely feeds data into the machine and does not know what weights will result after the training.

However, it is instructive to consider that similar arguments were raised when photographs were first considered for copyright protection. It was asserted that it was the camera that 'authored' the picture since the photographer merely pushed a button. But the Supreme Court rejected this argument and held that a camera is just a tool, and the selection of various factors such as the subject, the lighting, the viewing angle, distance, etc., comprised sufficient 'intellectual invention' to meet the de minimus human authorship requirement (Wenskay 1991, 27).

Similarly, a neural computer might be viewed as merely a tool in which the creation and selection of the training data satisfy the requirements of original human authorship.

Both present-day computer hardware and software can be used in varying proportions to implement neural networks

(Johnson-Laird 1990). Software simulations, while relatively easy to program, tend to be too slow for practical applications, since they rely on the sequential operation of a conventional digital computer. Hardware neural systems, by contrast, are faster as they take better advantage of the parallelism inherent in neural networks (Wenskay 1991).

Neural networks embodied in conventional software would likely be as copyrightable as are other forms of software. The object code in conventional software consists of a set of numbers (zeros and ones) expressing a program in machine language. This code has been held to be copyrightable (Apple Computer v. Franklin Computer Corp., 1983). By analogy it would seem that neural computer 'programs' consisting of numbers representing weighted connections would also be copyrightable. As to hardware-based neural networks, courts have also held that object code fixed in hardware (ROM) is protectible (Midway v. Artic Int'l., Inc., 547 F. Supp. 999, aff'd, 704 F. 2d 1009, 7th Cir. 1983). One commentator suggests that in addition to the copyrightability of neural weights by analogy to object code, training facts might be copyrightable by analogy to source code (Johnson-Laird 1990).

To be protected by copyright law as it stands today, neural weights would also need to satisfy the statutory definition of a computer program as "a set of statements or

instructions to be used directly or indirectly in a computer to bring about a certain result" (17 U.S.C. sec. 101).

Although neural weights, given their random character, might not seem to be 'statements' or 'instructions,' the Copyright Act reveals an intention to be flexible enough to adapt to new technologies. For example, 17 U.S.C. @ 102 states that "copyright protection subsists...in original works of authorship fixed in any tangible medium of expression, now known or later developed".

Federal court decisions establishing the copyrightability of computer software may well have paved the way for the copyrightability of neural network weights. Some modification of the statutory law may be needed, particularly as to the extent to which a work can be authored by a computer and still qualify for protection.

CHAPTER FOUR

POLICIES UNDERLYING THE JUDICIAL DECISIONSThe Interests Being Protected

The Federal courts have been accused of rendering software copyright decisions absent a clear theoretical understanding of the goals of copyright law. For example: "Without a clear vision of the end to be served by copyright protection, current judicial analyses of copyright problems drift aimlessly from the ad hoc to the arbitrary" (Spivack 1988, 777). While, hopefully, this is too sweeping an indictment, it is true that software programs are not easily analogized to any other form of authorship receiving copyright protection, and the precedential nature of our judicial system necessitates such incremental reasoning by analogy.

The ideal of copyright law has traditionally been to serve the interests of both innovators and the public. Some goals of such law articulated by the courts include: (1) providing sufficient legal protection in order to encourage creativity, (2) increasing the information, facts, and ideas in the public domain, (3) protecting investment in software development, (4) providing incentives for creators to share the results of their work so as to inspire further creative endeavors, (5) promoting a competitive market, and (6) increasing the information, facts, and ideas in the

public domain. Additionally, some commentators would like to add promotion of both standardization and efficiency to this list. The Federal courts have attempted to balance these sometimes competing interests while simultaneously rendering justice in the particular cases before them.

The purpose of intellectual property law stated in the United States Constitution is "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" (Art I, sec 8, cl 8). The goal is thus to encourage authorship, which may not be the same thing as encouraging authors.

When individual authors claim that they are entitled to incentives that would impoverish the milieu in which other authors must also work, we must guard against protecting authors at the expense of the enterprise of authorship (Litman 1990, 969).

The financial reward received by the author should be merely incidental to the general objective of encouraging innovation. As CONTU warned in 1978, "Copyright should not grant anyone more economic power than is necessary to achieve the incentive to create" (CONTU Final Report, 12).

A traditional purpose of the copyright system has been to leave as much information as possible in the public domain while protecting a creator's innovation from immediate piracy. This is accomplished by granting a limited monopoly to the author. It is limited by the right granted to others to, for example, make a backup copy of a

creator's computer program (17 U.S.C. sec. 117, 1982).

Also, in marked contrast to the ownership of most personal property or land, protection of intellectual property is limited to a fixed term.

Most copyrights on works created before 1978 last for twenty-eight years, with option to renew for an additional forty-seven; copyrights on works produced during or after 1978 last either for the author's lifetime plus fifty years or, in the case of work produced for hire [ie., for an employer], for seventy-five years from publication or 100 years from creation, whichever is shorter (Nance 1990, 758).

Limiting the monopoly granted to the programmer or to the corporate employer is an attempt to reward creativity without interfering with it. The problem is how to reward the efforts of today's innovators without requiring their future counterparts to 'reinvent the wheel'. The solution lies in leaving as much information as possible in the public domain while providing the minimum copyright protection needed to stimulate creation (Walter 1988, 8). Such protection should not go substantially beyond the purposes of copyright, since even a limited monopoly still burdens both competitors and the public. The author can, of course, choose to keep a copyrighted program off the market altogether.

Little useful information is asked of the software creator by the Copyright Office in the Library of Congress. Traditionally, when a work other than software is registered, copies are presented to the Library of Congress for indexing and public viewing or playing. In Ladd v. Law

and Technology Press (762 F. 2d 809, 9th Cir. 1985) a Federal appellate court ruled that this deposit requirement "directly further[ed] the purpose of promoting arts and sciences by adding to the collection of our national library" (814-15). Yet, because only a small fraction of the source code must be deposited along with the machine readable object code, information placed in the public domain is minimal.

The modest deposit requirement has worked to diminish society's historical archive. An important mission of the Library of Congress is to archive most examples of published works in the modern world. Yet, the public is denied access to a vast body of software in a form that can be read by human beings. This is leaving an enormous gap in such areas as the history of the early era of computers.

Economic Considerations

A fundamental assumption in a free market economy is that, absent imperfections in a market, competition alone will assure an efficient allocation of resources. Some economists find two imperfections which may justify government intervention in the market for computer software. Such intervention can take the form of patent or copyright protection, trade secret or unfair competition laws, mandatory licensing, or subsidies.

One market imperfection is referred to as the 'public

goods problem'. In economic terms, a 'public good' is one that has the property of nonexclusivity, ie., once the good has been produced, it is impossible or prohibitively expensive to exclude any individual from benefitting from it, whether or not he or she pays for it.

Indeed, individuals have an incentive not to pay for the good, or to undervalue it, in hopes of getting access as 'free riders'. The inability to exclude free riders distorts market signals and is thought to result in inefficient allocation of resources to nonexclusive goods and underproduction of them, relative to socially optimal quantities (Office of Technology Assessment 1992, 185).

Such goods are insufficiently produced because producers are unable to fully benefit from their investments in creating them.

CONTU recognized that the information comprising software innovation in computer programs is a public good and was unanimous in its belief that such programs were entitled to some form of legal protection "to encourage the creation and broad distribution of computer programs in a competitive market" (CONTU Final Report 1978, 11). Such protection was needed because (1) more powerful and less expensive computers had become widely available to individuals in homes and offices, (2) programs were no longer being written for single-purpose computers, and, (3) the cost of duplicating programs had become far less than the cost of creating them.

The public goods problem today is of less significance for application programs written for mainframe or

minicomputers than for microcomputers, since producers of the first two categories can recover their research and development costs through updating and maintenance contracts. They can minimize the problem of unauthorized copying through enforceable licensing provisions prohibiting disclosure and copying (Menell 1989, 38).

However, the public goods problem becomes acute with regard to the application code for microcomputers:

The ease and low cost of copying programs, the infeasibility of licensing restrictions, and the drawbacks to technological anticopying devices give rise to potential for significant 'leakage' of intellectual work to nonpurchasers (Menell 1989, 1079).

Few dispute the need to protect microcomputer program code against literal copying to prevent an imitator from immediately entering the first programmer's market much less burdened by product development costs. This would discourage innovation and reward waiting.

However, extending protection too far beyond prohibiting literal copying can create unwarranted monopolies which may adversely impact the software marketplace. One monopoly cost can be an intentional underproduction and overpricing of protected software. Another potential cost can be "unintended 'spillover' effects (e.g., transfer market power from the software market to the hardware market or vice-versa)" (Office of Technology Assessment 1992, 186). Additional monopoly costs can be the need to 'invent around' a program 'locked up' by

a copyright. There can also be the costs of prosecuting or defending infringement lawsuits.

Some of these monopoly costs might be justified when copyright protection encourages important innovation that would be slow in coming in its absence. "But when the principal effect of furnishing legal protection is to exclude others from building on a readily apparent outgrowth of existing technology, the costs of protection clearly outweigh its benefits" (Menell 1989, 1079). Thus far, both Congress and the courts have accepted the deleterious economic effects of monopoly in order to provide an incentive for innovation (Wright 1991).

Another market imperfection which may provide economic justification for government intervention in the marketplace by providing copyright protection is the presence of 'network externalities'. Network externalities exist in markets for products for which the user's satisfaction increases with the number of other consumers of the same product. In the software market such satisfaction results from the wider use of a particular user interface. "Computer-human interfaces generate network externalities to the extent that standardization produces larger networks and reduces retraining and mobility costs" (Menell 1989, 1095). Network externalities are much more important today than at the time of CONTU due to the larger size of today's software market.

Commentators are not in agreement as to whether the effect of intellectual property laws on standardization should affect judicial determinations of the appropriate level of copyright protection of user interfaces. One view is that protection can slow the standardization process by increasing the vested interest each firm has in holding back from cooperating in voluntary standards setting in the hope that its own interface will eventually become the industry standard. This would burden users who must learn new interfaces for different programs. Protection may also prevent new firms from entering a market where a larger firm's interface has almost become the de facto standard, yet is still protected by copyright.

An opposing view as to the need to consider standardization when evaluating copyright protection of the user interface is that standardization should be encouraged outside of intellectual property law through completely voluntary cooperation among software companies, user interface consortia, and standards-setting organizations (Office of Technology Assessment 1992).

Some favoring more government involvement would have the government intervene in favor of an interface once it achieves a wide enough market to be considered a de facto standard. The government could then "set standards directly, relax antitrust restraints that prevent private firms from setting voluntary standards, or promote

standardization through its market power as a major purchaser" (Menell 1987, 1344). The overriding issue is how to strengthen standards while simultaneously promoting competition and innovation.

Lotus Development Corporation v. Paperback Software International (1990) provides insight into how economic considerations come to be incorporated into Federal copyright cases. The court in that case was called upon "to determine whether Congress has extended copyright protection to nonliteral elements of computer programs, and if so, to what extent" (Lotus 1990, 46). In order to do this, the court examined (1) the actual language of the copyright statutes, (2) the broader statutory context (eg., how non-literal elements had been treated in musical works under analogous statutory provisions), and, (3) the 'object and policy' of copyright law. It is while seeking the object and policy of the law that the court looked to Peter S. Menell's article, "An Analysis of the Scope of Copyright Protection for Application Programs" (1989), which it quoted as follows:

Drawing the line too liberally in favor of copyright protection would bestow strong monopolies over specific applications upon the first to write programs performing those applications and would thereby inhibit other creators from developing improved products. Drawing the line too conservatively would allow programmer's efforts to be copied easily, thus discouraging the creation of all but modest incremental advances (Lotus 1990, 52 and Menell 1989, 1047).

Although the court quotes Menell's article, it does not

adopt the author's thesis, which is that the idea-expression merger doctrine should be applied by the courts so as to extend copyright protection beyond literal copying only when the economic benefits are likely to exceed the costs.

The linkage between copyright protection and economic benefits to society as a whole has traditionally followed this logic: copyright protection increases innovators' ability to obtain returns from their intellectual labors; the resulting potential for increased private gains to innovators stimulates additional innovation; and, because of increased innovation, additional benefits accrue to society as a whole (Office of Technology Assessment 1992). In Mazer v. Stein (347 U.S. 201, 219, 1954) the U.S. Supreme Court stated this rationale as follows:

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 (347 U.S. 201, 219).

While this economic philosophy of translating the individual profit motive into a public good has strong basis in judicial precedent, an examination of economic costs does not. The court in Lotus did not look at such potential economic costs of protecting the Lotus Development Corporation's copyright as (1) the need for other companies to engage in otherwise unnecessary research and development because useful information has been removed from the public domain, (2) the administrative costs of licensing

innovations from the copyright holder, (3) social costs of using the legal system to enforce copyright rights, and, (4) the foregone benefits to society from other innovations which might have taken place absent copyright protection.

The courts have been influenced by a pervasive assumption that, from an economic perspective, stronger copyright protection will increase private incentives for innovation, thereby producing additional technological progress and increased benefit to society overall (Breyer 1970). But this is not necessarily the case. Theoretical and empirical work from the mid-1970's to the present has challenged the assumption that stronger intellectual property protection yields positive economic consequences. "[T]here is now no general theoretical presumption in economics that the consequences of 'stronger' protections are favorable for society as a whole" (Office of Technology Assessment 1992, 187). If, indeed, the courts are affording stronger copyright protection than necessary to generate an optimal level of research, development, and marketing, this may represent an undue burden on society.

CHAPTER FIVE

CONGRESSIONAL OPTIONS REGARDING SOFTWARE PROTECTIONGeneral Considerations

Copyright has emerged as the predominant form of legal protection for both application program code and for the user interface. Given the potential negative social costs of such protection, a copyright should not provide any person or organization more economic power than is needed to achieve the incentive to create. Copyright law should set a balance between (1) providing just the needed degree of proprietary protection to stimulate innovation, and, (2) leaving enough information in the public domain so as not to stifle independent creation by others. This delicate balance can be disrupted by changes in (1) the software industry, (2) software technology, or (3) copyright law itself.

Since Congress created the CONTU commission in 1974, an explosive growth in personal computers and packaged software has transformed the software industry. Small networks of software hobbyists of the 1970's have given way to an enormous mass market, and the industry is now dominated by about 200 companies, including some hardware giants such as IBM. Larger firm size has meant increased barriers to entry by smaller firms. Such barriers include expensive start-up costs and an inability to attract capital because of the

lack of clear legal protection for newly-developed software (Hooten 1990). Even if such barriers are overcome, the smaller firm's capital investment may be unduly jeopardized by infringement litigation or the threat of such litigation.

"[P]rotections that were once sought against 'piracy' have come to be used by major players as instruments of competition" (Rice 1990, 350). This can limit competition as individuals and companies become far more cautious before introducing works into the marketplace.

In addition to changes in the software industry that may threaten the balance between stimulating innovation and fostering free access to information, a less quantifiable change has occurred in copyright law itself. Apple Computer v. Franklin Computer (1983) and other early cases established copyright protection against literal or verbatim copying which directly undercut the market. These early cases "reached both user, or vertical market, and competitor, or horizontal market, copying of marketable computer programs" (Rice 1990, 354). Later cases, such as Whelan (1986) and Broderbund (1986) were more expansively protective, and enhanced protection against primarily horizontal market competition.

In a market economy, however, that is precisely the realm in which it is most important to reject...the use of copyright to secure exclusive economic rights to the functional attributes of computer programs. Traditional copyright law recognizes that. If anything, respect of the principle and its purpose is even more important in the case of what are truly technological, not authentically literary, works (Rice 1990, 354).

Despite such criticism of perhaps overly zealous judicial protection of horizontal markets, however, court decisions alone should not be held solely responsible for the current turf wars between software competitors. The Federal courts have had to attempt to resolve specific disputes within a given, ill-fitting copyright framework. An important example of this poor fit was the necessity of classifying computer programs as literary works in order to find some established niche for software within existing categories of copyright law. But such classification meant that protection of the structure and organization of software was compelled by the already-established protection of these elements in traditional literary works. This protection has led to the current wave of 'look and feel' lawsuits, which some believe are undermining public confidence in the ability of the judicial system to provide other than ad hoc software copyright protection.

Dissatisfaction with current law does seem to be most acute with respect to copyright protection of a program's 'look-and-feel'. For example, a May 2, 1989 survey of 667 interface researchers, designers, and developers attending the sixth Association for Computing Machinery convention, found that 80 percent "opposed copyright protection for the look and feel of user interfaces, although they strongly supported copyright protection for source and object code" (Samuelson 1990, 484). Although this informal survey did

not consider the size of the respondents' employing firms, it illustrates the widening gap between the legal precedents being set and the behavior of many in the industry.

Maintaining the Status Quo

Absent new legislation, the courts will continue to render decisions based upon current statutes. Caselaw will continue to evolve, as in the past, by judicial decisions interpreting the law while often looking to the legislative intent of CONTU.

Those favoring this status quo prefer to rely upon gradual change brought about judicially rather than a sudden alteration of copyright law by Congress. They have faith in a system in which they believe many courts analyzing disparate factual situations in the same field of law can develop a more comprehensive and just collection of policies than would any one body attempting to legislate without benefit of such close contact with the law's application.

Each discrete judicial decision is binding precedent for other cases only on the same facts. If a particular court makes a mistake, other courts can distinguish their facts and correct the mistake. From many narrow decisions comes one broad policy (Davidson 1986, 1067).

Uncertainties and delay are, arguably, the price to be paid for incrementally implementing a system in which a more accurate policy can be developed, since "no one person, or group of persons, is bright enough, farsighted enough and wise enough to integrate the complexity of a new technology

like software into our economy" (Davidson 1986, 28). The system is also protected from special interest lobbying for legislation favoring one commercial group or another.

Those opposed to allowing the common law process alone to determine policy regarding intellectual property see the resulting uncertainty as too high a price to pay. They point to the divergent positions of the Federal trial and appellate courts in different jurisdictions and to unwarranted copyright monopolies wielded by a few large software giants as undermining the pace of innovation needed for global competitiveness.

How Comprehensively to Act

CONTU realized that its efforts were only a beginning towards defining the scope of software protection. The Commission recommended, therefore, that Congress should periodically review this protection. This "should be performed well and often enough to prevent the copyright law from becoming as anachronistic as did the 1909 Act" (CONTU Final Report 1978, 12).

In May, 1992, the Office of Technology Assessment (OTA), comprised of six members of the House and six members of the Senate, issued a report finding that although incremental judicial accommodation to changes in the software industry since CONTU has advantages, the

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 (OTA 1992, 26, emphasis in the original).

Should Congress decide to act, Federal statutory changes could range anywhere from modest procedural adjustments to a completely new act to replace current copyright and patent law. Intermediate approaches include the creation of one or more 'sui generis' protections. 'Sui generis' is a Latin phrase describing a law that is 'of its own class'. A sui generis system would remove software, or certain aspects of software, from the realm of copyright and patent law and fashion a new system of protection.

It is significant that the OTA is willing to suggest a sui generis treatment of software despite the difficulties this may present for reciprocal international copyright agreements and the discontinuity this would inject into the common law process.

Computer chips have already been singled out for such special treatment in the Semiconductor Chip Protection Act of 1984 (17 U.S.C. 900). This Act constituted a recognition that pre-existing categories were inadequate to deal with a new technology. Rather than forcing copyright law to include that which it was never intended to protect, the new system attempted to adapt a new form of protection to a new technology. Thus far, software has not been singled out for such special treatment, partly because the diversity of software itself, as compared to semiconductor chips, would

make special legislation difficult to formulate.

Some argue, however, that given the poor fit of software within the copyright framework, such special legislation is essential to minimize inefficiencies in software development, to promote standardization, and to restore public confidence in the principle of intellectual property rights.

Numerous suggestions have been made as to what form a sui generis system might take. Most commentators favor a modified copyright approach (Office of Technology Assessment 1992, 27). For example, adjustments could be made in the copyright registration requirements. Fuller disclosure could be achieved by requiring that the complete software source code in human readable form be placed on file at the Copyright Office, rather than merely enough source code to identify the machine readable object code currently required (Mitchell 1991, 463).

Software producers realize most of their return from research and development within a short time after they market their products (Menell 1987, 1329 n. 206). This, along with a marketable life span of about ten years for most programs, has led some to recommend a much shorter term of protection than that available under traditional copyright law. Ten years is a recurring suggestion (eg., Samuelson 1984, 766; Antton and Hoffman 190, 9; Mitchell 1991, 463). The time period selected would need to take

into account the erosion of lead time advantages arising from computer-aided software development (Office of Technology Assessment 1992, 10).

One suggestion for re-establishing greater balance in protecting both the creator and the public would be to protect against copying and not against independent development by making 'value added' to an appropriated element a complete defense to infringement.

Such a law would result in liability for parties like Paperback Software, but would spare innovators such as Borland International (publishers of the 'Quattro' and 'Quattro Pro' spreadsheets) because of the value they added to the appropriated elements (Antton and Hoffman 1990, 10).

A frequent suggestion is to require mandatory licensing, perhaps a certain time period after registration, in order to provide access by other potential innovators.

Absent some form of systemic adjustments, the early warnings of CONTU commissioners John Hersey and Melville Nimmer may yet be realized. Dissenting Commissioner Hersey stated that a "forceable wrenching" would be required to fit computer programs in machine-readable form into the copyright scheme. Commissioner Nimmer advised that adoption of the Commission's recommendations might take copyright law "beyond the breaking point" (CONTU Final Report 1978, 26-7).

CONTU's Final Report recommended only modest changes in the 1976 Copyright Act so that the full body of copyright law would apply to computer programs. The relatively short history of trying to fit software into traditional copyright

law appears to have created an imbalance in favor of software producers to the detriment of competitors and the public.

Congress should appoint a second National Commission on New Technological Uses of Copyrighted Works comprised of computer scientists, industry representatives, professional programmers, software industry and hardware industry representatives, large users and small users of computer systems, economists, and attorneys. The Commission should assess a broad range of policy options while taking into account the likely impact of software copyright law on newly emerging technologies such as neural networks and DNA.

The Commission should sponsor hearings nationwide, both to maximize public input and to reduce the risk that one group might disproportionately influence any resulting legislation. Topics to be addressed should include (1) the economic and social costs borne by society under the current level of software copyright protection; (2) the extent to which innovation is being unduly inhibited by current law; (3) the relationship between copyright law and software standardization; (4) the need to change copyright registration to require full disclosure of the source code text, perhaps in stages so as to allow time for creators to gain from their innovations before competitors have full access to such code; (5) the possibility of shortening the term of copyright protection; and, (6) the extent to which a

work authored by a computer, rather than by a human being, should still qualify for some form of copyright protection.

Absent further statutory guidance in these areas, the Federal courts will be left to apply the Congressionally mandated idea-expression distinction with insufficient information as to what the object and policy of copyright law should be in the information age. Legislative clarification in the form of a carefully crafted statute could serve to restore the delicate balance between the incentive to innovate and public access to information.

A new statute could enable corporate software rivals to build upon one another's advances rather than litigating to block their competition. By enhancing the conditions under which new technologies can flourish, copyright law can enlarge the pool of knowledge necessary for the development of the software industry so that the public benefit is maximized.

CASES CITED

Cases Cited

- Apple Computer, Inc. v. Microsoft Corporation and Hewlett-Packard Corporation, 759 F. Supp. 1444 (N.D. Cal. 6 March, 1991).
- Apple Computer, Inc. v. Microsoft Corporation and Hewlett-Packard Corporation, 779 F. Supp. 133 (N.D. Cal. 13 August, 1991).
- Apple Computer, Inc. v. Franklin Computer Corporation, 545 F. Supp. 812 (E.D. Pa. 1982).
- Apple Computer, Inc. v. Franklin Computer Corporation, 714 F. 2d 1240, (3rd Cir. 1983).
- Broderbund Software, Inc. and Pixellite Software v. Unison World, Inc., 648 F. Supp 1127 (N.D. Calif. 1986).
- Brown Bag Software, Inc. v. Symantec Corporation and Softworks Development, Inc., No. 89-16239 (9th Cir. April 7, 1992).
- Data Cash Sys., Inc. v. JS&A Group, Inc., 480 F. Supp. 1063 (N.D. Ill. 1979).
- Donaldson v. Beckett, 4 Burrow's Reports 248 (1774).
- Herbert Rosenthal Jewelry Corporation v. Kalpakian, 446 F. 2d 738 (9th Cir. 1971).
- Ladd v. Law and Technology Press, 762 F. 2d 809 (9th Cir. 1985).
- Lotus Development Corporation v. Paperback Software International, 740 F. Supp. 37 (D. Mass. 1990).
- Mazar v. Stein, 347 U.S. 201 (1954).
- Midway v. Artic Int'l., Inc. 547 F. Supp. 999, aff'd., 704 F. 2d 1009 (7th Cir. 1983).
- National Comics Publication v. Fawcett Publications, 191 F. 2d 594 (2d Cir. 1951).
- Nicholas v. Universal Pictures Corporation, 45 F. 2d 119 (2d Cir. 1930).
- Q-CO Industries, Inc. v. Sidney Hoffman, Dilip Som and Computer Prompting Corporation, 625 F. Supp 608 (S.D. N.Y. 1985).

Synercom Technology, Inc. v. University Computing Co. and Engineering Dynamics, Inc., 462 F. Supp. 1003 (N.D. Texas 1978).

Twentieth Century-Fox Film Corporation v. MCA, Inc., 715 F. 2d 1327 (9th Cir. 1983).

Whelan Associates v. Jaslow Dental Laboratory, 797 F. 2d 1222 (3rd Cir. 1986).

REFERENCE LIST

Reference List

- Adelstein, Richard P. and Steven I. Peretz. 1985. The competition of technologies in markets for ideas: Copyright and fair use in evolutionary perspective. International Review of Law and Economics 5 (December): 209-238.
- Antton, D. Lee and Gary M. Hoffman. 1990. Copyright protection and innovation: The impact of Lotus Development v. Paperback Software. The Computer Lawyer 7 (August): 1-11.
- Bender, David. 1991. Computer law: Software protection. New York: Times Mirror Books.
- Borden, Barry. 1983. The Apple-Franklin Decision: It will protect entrepreneurs and help America stay ahead in computers. "Los Angeles Daily Journal", 29 September, p.4 (col. 3).
- Branscomb, Anne W. 1988. Who owns creativity? Technology Review 91: 38-45.
- Breyer, Stephen. 1970. The uneasy case for copyright: A study of copyright in books, photocopies, and computer programs. Harvard Law Review 84 (December): 281-351.
- Carter, Stephen L. 1990. Owning what doesn't exist. Harvard Journal of Law and Public Policy 13:99-107.
- Chaffee, Zechariah Jr. 1945. Reflections on the law of copyright: I. Columbia Law Review 45 (July): 503-529.
- Clapes, Anthony L., et. al. 1987. Silicon epics and binary bards: Determining the proper scope of copyright protection for computer programs. U.C.L.A. Law Review 34 (June-August): 1493-1594.
- Clapes, Anthony L. 1984. Software, copyright and competition: The 'look and feel' of the law. New York: Quorum Books.
- Cruz, Amaury. 1990. What's the big idea behind the idea-expression dichotomy? -- Modern ramifications of the tree of porphyry in copyright law. Florida State University Law Review 18 (Summer):221-250.
- Daly, James. 1992. Suit may hurt Microsoft users. Computerworld 37.

- Davidson, Duncan M. 1986. The future of software protection: Common law, uncommon software. University of Pittsburgh Law Review 47 (Summer): 1037-1073.
- Drysdale, David B. 1987. Thoughts on copyright: computer programs. Computer Law and Practice 3 (January/February): 91-98.
- Erstling, Jay A. 1989. The Semiconductor Chip Protection Act and its impact on the international protection of chip designs. Rutgers Computer and Technology Law Journal 15: 303-349.
- Fetterman, Daniel J. 1986. The scope of copyright protection for computer software: Exploring the idea/expression dichotomy. Copyright Law Symposium, No. 33. New York: Columbia University Press: 1-32.
- Francis, Robert. 1988. Apple-Microsoft suit raises issue of vague copyright laws. Datamation 15 (May): 22-26.
- Gates, William H. 1983. The Apple-Franklin decision. Los Angeles Daily Journal, 29 September, 4.
- Giannetti, Thomas. 1990. Software protection expanded by controversial Lotus decision. The National Law Journal, 6 August, 39-44.
- Gomes, Lee. 1992. An 'instant bounce' machine: Device knows if check is good. San Jose Mercury News, 4 June, E1, E6.
- Gomes, Lee. 1992. An 'instant bounce' machine: Device knows if check's good. "The San Jose Mercury News", 4 June, 1E.
- Groenewold, Glenn. 1988. A poisoned apple; Apple computer's 'look-and-feel' copyright lawsuit. UNIX Review 6 (December): 36-39.
- Halvey, John K. 1988. A rose by any other name: Computer programs and the idea-expression distinction. American Society of Composers, Authors, and Publishers. Copyright Law Symposium No. 35. New York: Columbia University Press: 1-40.
- Hamilton, Alexander, et. al. 1961. The Federalist Papers. New York: Times Mirror.
- Hoffman, Gary M. and John D. Grossman. 1990. Substantial similarity at issue in software infringement cases. The National Law Journal, 5 March, 34.

- Hooten, Karen. 1990. The fuzzy logic of intellectual property: Programming by profession. Computer Language 7 (September): 117-121.
- Illinois Legal Times Roundtable. 1991. Illinois Legal Times, October, 1.
- Johnson, Dawn E. 1990. Computer software: Enforceable protection versus the free use of ideas. Western State University Law Review (Spring): 373-414.
- Johnson, Brian. 1991. An analysis of the copyrightability of the 'look and feel' of a computer program. Ohio State Law Journal 52 (Summer): 947-989.
- Johnson-Laird, Andy. 1990. Neural networks: The next intellectual property nightmare? The Computer Lawyer 7 (March): 7-22.
- Kaplan, Lawrence M. 1988. Screen displays are proper subject matter for copyright protection. University of Illinois Law Review 1988: 757-787.
- Kehoe, Louise. 1992. Judge to review Apple copyright decision. Financial Times, May 14, 1992, 27.
- Kidwell, John A. 1985. The Semiconductor Chip Protection Act of 1984 and its lessons: Software and semiconductors: Why are we confused? Minnesota Law Review 70 (December): 533-576.
- Kostal, Susan. 1992. After Apple, 'look-and-feel' seems less than it once was. San Francisco Daily Journal, 16 April, 1.
- KQED National Public Radio. 1992. Forum, 16 April.
- Leaffer, Marshall A. 1989. Understanding Copyright Law. New York: Matthew Bender.
- Lewis, Gerard J. 1991. Lotus Development Corp. v. Paperback Software International: Broad copyright protection for user interfaces ignores the software industry's trend toward standardization. University of Pittsburgh Law Review 52 (Spring): 689-720.
- Litman, Jessica. 1990. The public domain. Emory Law Journal 39 (Fall): 965-1023.
- Mackaay, Ejan. 1990. Economic incentives in markets for information and innovation. Harvard Journal of Law and Public Policy 13 (Summer): 867-909.

- Meisner, Mary M. 1988. Archival backup copying of software: How broad a right? Rutgers Computer and Technology Law Journal 14: 391-407.
- Menell, Peter S. 1989. An analysis of the scope of copyright protection for application programs. Stanford Law Review 41 (May): 1045-1103.
- Menell, Peter S. 1987. Tailoring legal protection for computer software. Stanford Law Review 39 (July): 1329-1372.
- Mitchell, Christine. 1991. Copyright: The great debate. The Law Institute Journal 65 (June): 462-463.
- Moreno, Richard D. 1990. 'Look and feel' as a copyrightable element: The legacy of Whelan v. Jaslow: Or, can equity in computer program infringement cases be found instead by the proper allocation of burden of persuasion? Louisiana Law Review. 51 (September): 177-215.
- Nance, Dale A. 1990. Forward: Owning ideas. Harvard Journal of Law and Public Policy 13 (Summer): 758-773.
- Nasri, William Z. 1976. History of copyright. New York: Marcel Dekker, Inc.
- National Commission on New Technological Uses of Copyrighted Works (CONTU). 1978. Final Report of the National Commission on New Technological Uses of Copyrighted Works. Washington, D.C.: U.S. Government Printing Office.
- Neitzke, Frederic W. 1984. A software law primer. New York: Van Nostrand Reinhold Co.
- News Services. 1992. Microsoft splits its stock 3-for-2 as outlook zooms. The Atlanta Journal and Constitution, 28 April, 5 (C).
- Office of Technology Assessment (OTA). 1992. Finding a balance: Computer software, intellectual property and the challenge of technological change. Washington, D.C.: U.S. Government Printing Office.
- Orenstein, Susan. 1992. Judge takes bite from Apple's copyright suit against Microsoft. The Recorder, 15 April, 2.
- Orenstein, Susan. 1992. Vexed Walker gives Apple another chance. The Recorder, 13 May, 1.

- Orenstein, Susan. 1992. Why Apple lost Microsoft gamble. The Recorder, 17 April, 1.
- Orenstein, Susan. 1992. Apologetic Apple renews 'look and feel' claim. The Recorder, 12 May, 3.
- Pilarski, John H. 1990. User interfaces and the idea-expression dichotomy or Are the copyright laws user friendly? American Society of Composers, Authors, and Publishers. Copyright Law Symposium No. 37. New York: Columbia University Press: 45-83.
- Quick, Gregory. 1992. Microsoft-Apple infringement suit rumbles on. Computer Reseller News, 24 February, 16.
- Rice, David A. 1990. Whither (No longer whether) software copyright. Rutgers Computer and Technology Law Journal 16 (Winter): 341-355.
- Robinson, Gerald H. 1990. Protection of intellectual property in neural networks. The Computer Lawyer 7 (March): 17-23.
- Samuelson, Pamela. 1984. CONTU Revisited: The case against copyright protection for computer programs in machine-readable form. Duke Law Journal 1984: 663-769.
- Samuelson, Pamela and Robert J. Glushko. 1990. Survey on the look and feel of lawsuits. Communications of the ACM 33 (May): 483-497.
- Shelly, Gary B., et. al. 1990. Computer concepts with microcomputer applications. New York: Boyd and Fraser Publishing Company.
- Smedinghoff, Thomas J. 1990. Developments in software copyright law. Software Law Journal 3: 637-686.
- Spivack, Peter G. 1988. Does form follow function? The idea/expression dichotomy in copyright protection of computer software. UCLA Law Review 35 (April): 723-762.
- Stanley, Steven M. 1981. The new evolutionary timetable: Fossils, genes, and the origin of species. New York: Basic Books.
- Stern, Richard H. 1986. The future of software protection: The bundle of rights suited to new technology. University of Pittsburgh Law Review 47 (Summer): 1229-1267.

- Stokes, Michael D. 1985. Copyright law and integrated circuit protection: When the chips are down. Computer Law Journal 6 (1985-6): 543-569.
- Walter, Charles. 1988. Defining the scope of software copyright protection for maximum public benefit. Rutgers Computer and Technology Law Journal 14: 1-156.
- Wenskay, Donald L., et. al. 1991. Neuural networks: A prescription for effective protection. The Computer Lawyer 8 (August): 12-30.
- Wright, William H. 1991. Litigation as a mechanism for inefficiency in software copyright law. U.C.L.A. Law Review 39 (December): 397-436.
- Zachmann, William. 1992. Apple lawsuit threatens to cost Microsoft plenty. PC Week Opinion, 17 February, 74.