*Just When We Thought "Things Never Change" -- Will Lightening Strike Twice?**

Ronald L. Johnston
Arnold & Porter
Los Angeles

Ronald Johnston has served as a Responsible Partner for Arnold & Porter's Intellectual Property Group and its Los Angeles office. He has represented clients in leading cases in the information technology and entertainment industries, both at the trial and appellate levels, for more than 30 years. This representation has included matters concerning the scope of intellectual property protection for computer software, including software databases, languages, application programming interfaces, and external and internal interfaces, as well as the application of unfair competition and antitrust laws to computer software and technology products and services.

Mr. Johnston co-founded and serves as Editor-in-Chief of *The Computer & Internet Lawyer* (1984-present), a leading journal on Internet law, and was the founder and continuing Chairperson of the University of Southern California's Computer & Internet Law Institute (1979-2003). He has been featured in *LA Magazine* as a "trailblazer of the law," and recognized in the U.S. and International editions of *Chambers* (for Litigation and IP), the *LA Business Journal* as "the best of the bar," and Best Lawyers as Los Angeles "Information Technology Law Lawyer of the Year for 2012." The *California Daily Journal* has named Mr. Johnston among the "Top 100 Lawyers in California" and the "Top 75 Intellectual Property Attorneys." In recognizing Mr. Johnston, the *Daily Journal* observed: "More than 30 years after he imbedded himself at the intersection of the law and technology, Mr. Johnston has remained remarkably relevant in an industry that has a penchant for reinventing itself at an astonishing pace," and he has been "instrumental in shaping some of the biggest policy decisions regarding the ever-expanding digital frontier."

*Any views expressed in these materials are only those of the author individually -- if that.

TABLE OF CONTENTS

* * *

Most of the blockbuster reported decisions on the scope of copyright protection for computer software ended in the 1990s. At that time, there were open issues regarding the basics of protection for non-literal elements of software, such as important unanswered questions regarding the level of abstraction at which software may be protectable; how you determine what is non-protectable idea/system/method of operation versus protectable expression; whether computer languages or command structures are subject to copyright protection; how to compare two programs to determine infringement, including what level of similarity must exist and how do you consider elements of compilation; what significance attaches to considerations of compatibility or interoperability; whether fair use might preempt contractual restrictions on reverse engineering/disassembly and/or trade secret protection for source code; and so on. Indeed, the Fifth Circuit characterized the state of case law as of 1994 as "Court decisions are, *generously stated*, in a state of *creative ferment* concerning the methods by which nonliteral elements of computer programs may be identified and analyzed for copyrightability." *Engineering Dynamics v. Structural Software*, 26 F.3d 1335, 1340 (emphasis added).

Nonetheless, the answers to these issues largely stopped coming some time in the 1990s. As a result, the issues regarding the scope of copyright protection for software outstanding today have not really changed much since the turn of the century. Professor Nimmer's treatise today states: "Software developers have no adequate guidelines regarding what level of independent development is required to avoid copyright infringement [fn. 83]." All of the cases cited by Prof. Nimmer today in footnote 83 were decided in the 1980s!

Then came the Federal Circuit's decision in *Oracle v. Google*. It tackles head on issues of copyright protection for command and application interfaces left relatively untouched since the First Circuit's decision in 1995 in *Lotus v. Borland*, 49 F.3d 807. Rejecting the First Circuit approach, the Federal Circuit held that application programming interfaces ("APIs"), including the command language and syntax, and organization, structure and sequence of the APIs, may be appropriate subject matter for copyright protection. The case remains pending to determine the applicability of the fair use defense and likely will have a future appellate track, including potentially back to the Supreme Court.

With the many other issues regarding the scope of protection for software outstanding, might lightning strike twice? Might other parties bring to court potential disputes or ambiguous relationships arising out of the relatively basic issues regarding the scope of copyright ownership of software that have been outstanding for 20 or 30 years? Section II of these materials identifies some of the issues that remain outstanding from fairly early in the history of the software industry.

I. Some Things May Have Changed: Oracle v. Google

A. The Year 1991 -- In Review

In 1991, I was asked to participate in a mock argument on the protectabiliy of languages and other elements of user interfaces. Below is my written submission, based on a brief I had authored on such issues (in a case that was resolved by settlement).

The setting for the written submission is the time between the district and appellate opinions in *Lotus v. Borland,* the latter of which recently was rejected in *Oracle v. Google* and has never received universal acceptance. Based on the "hypothetical," the written submission flows from arguments regarding the protection of user interfaces generally to language protection.

The open issues haven't really changed much since 1991, as the writing below illustrates. Many of the conclusions reached by the Federal Circuit in the *Oracle* case are argued in the written submission below.

[From a Presentation at the University of Southern California Law School's Computer & Internet Law Institute]

COPYRIGHT PROTECTION FOR COMMAND DRIVEN INTERFACES

By Ronald L Johnston and Allen R. Grogan

This article presents the case for copyright protection for command driven interfaces, including protection for the command set or "language" of a computer program. The subject is addressed as it might be considered by a federal court, based on existing principles of copyright law. Issues regarding the scope of protection are analyzed in the context of programs that perform interactive data management tasks, such as Lotus 1-2-3, the dBASE programs, and many other popular software programs.

This article does not explicitly address public policy arguments such as might be made in academic literature or to a legislative body considering amendments to the copyright laws. A growing body of literature — but scant systematic analysis — discusses elsewhere alleged social and economic implications of copyright protection for software.

Opening Statement

Since the adoption of the Copyright Act of 1909, courts have repeatedly protected against copying visual images, sequencing and order reflected in original authorship, and compilations of words and phrases, regardless of the form of the work. Although copyright protection for computer programs has only been the subject of reported decisions during the last ten years, numerous courts have addressed the copyrightability of computer programs during this period. These courts consistently have extended copyright protection to every facet of a computer program, based on principles of copyright law established in non-software cases. As courts deciding software cases have recognized, the principles underlying the protection of other forms of authorship apply with equal force to the user interface of a computer program. The decisions in both software and non-software cases support copyright protection for the sequence or flow of screens, individual screen content, and the command set of a computer program.

I. THE USER INTERFACE OF A COMPUTER PROGRAM

A. The User Interface is a Highly Structured "Dialogue" Between the User and the Program.

The "user interface" of a computer program is the way in which the program presents itself to the user and includes the sequence of visual images that appears on the screen of the computer's monitor. The user interface consists of the interaction or dialogue between the user and the computer program as the computer performs the user's tasks. User interfaces vary substantially from one independently developed program to another.

The user interface determines the "look and feel" of the computer program to the user.[1] The user interface communicates information to assist the user in the operation of the program, such as information designed to lead the user through the steps necessary to accomplish the user's task and identifying those commands that the user must give the program in order to take each step. The user interface also commonly presents selected information regarding the status of the computer program's operation and presents data to the user in an organized and easy to understand format.[2]

---

[1] For example, the District Court for the Northern District of California gave this description of the user interface of a computer program in *Telemarketing Resources v. Symantec, Copyright L. Dec.* (CCH) ¶ 26,514 (1989):

> The user interface, also called the "look and feel" of the program, is generally the design of the video screen and the manner in which information is presented to the user.

[2] The district court in *Manufactures Technologies, Inc. v. Cams, Inc.*, 706 F. Supp. 984, 993 (D. Conn. 1989), offered this description of the function of the user interface in that case: "the user interface is designed to communicate with the user in a way to facilitate the understanding and use of the program itself."

For example, the user interface may lead the user, screen by screen, through the steps necessary to create, modify or query the database. The sequence or flow of screens to create the database might include: screen 1-- selecting a file name and designating the categories of information to be included in the database (*e.g.*, in a travel agent's database, customer name, customer address, vacation destination, departure date, price); screen 2 -- specifying the type of data to be entered (*e.g.*, character, numeric, logical); screen 3 -- specifying the number of characters that will be allowed for certain data to be entered, such as names; screen 4 -- inputting data regarding particular customers; and screen 5 -- editing or adding additional customers or "records" to the database.

The individual screens of the user interface may present distinctive content to the user. Such screens may include menus listing selected commands the program will then accept. Certain screens also may advise the user of matters such as the status of the operation of the program (*e.g.*, how many records have been created, where the file being created is stored), or the format of the data file being created.

Finally, through predefined sequences of commands to which the program will respond, the user, prompted by sequences of screens, directs the program to perform the steps necessary to create, modify or query the database. For example, the user might use a CREATE command to create a data file; BROWSE command to display the data for memory and editing; DISPLAY STRUCTURE command to instruct the program to display information about how the data is organized; DISPLAY STATUS command to display information regarding the data files currently in use and the settings of various options within the program; and APPEND command to add new records.

As illustrated by these examples, the most tangible expression of the ideas embodied in a computer program is the design of the user interface. It is the most communicative and informational facet of a computer program, and the only element of the program capable of communicating directly with or having aesthetic appeal to the program user. The design of the user interface of a program is an intellectual task, requiring original authorship by the program developers.[3]

B. The Elements of the User Interface

[3] The district court in *Lotus Development Corp. v. Paperback Software Int'l*, 740 F. Supp. 37, 56 (D. Mass. 1990), found:

> [T]he bulk of the creative work is in the conceptualization of a computer program and its user interface, rather than in its encoding, and that creating a suitable user interface is a more difficult intellectual task, requiring greater creativity, originality, and insight, than converting the user interface design into instructions to the machine.

The user interface of a computer program includes the following elements. First, the user interface consists of the specific sequence or flow of screens presented to the user. This sequence or flow of screens expresses the program developer's approach to the problem the user wishes to solve. *See Manufacturers Technologies, Inc. v. Cams, Inc., et al.*, 706 F. Supp. 984, 994 (D. Conn. 1989) (illustrating how a sequence of screens in a data base program communicates with the user in a predetermined dialogue).

Second, the user interface consists of unique individual screens. The individual screens convey information to the user of various types and prompt the user to take appropriate actions. Such screens may be expressive and original. *See Broderbund Software; Inc. v. Unison World, Inc.*, 648 F. Supp. 1127, 1134 (ND. Cal. 1986) (illustrating how terminology expressed on specific screens may be distinctive and informative); *Manufacturers Technologies*, 706 F. Supp. at 996.

Third, the user interface consists of the command set or command structure of the program. These commands are declarative statements or instructions through which the user communicates with the computer system. Section I.C., *infra*.

Although most computer programs that manage data perform certain generic functions or tasks (such as creating a database, adding new files and deleting files), each program varies substantially in how it accomplishes these tasks. These differences find expression in each of the elements of the user interface, including: the specific sequence or flow of screens the user interface presents to the user to accomplish a particular task; the organization, format and content of individual screens; and the command set, including the selection and sequence of commands the program will recognize, the terms chosen to express the commands, and the required syntax of command statements. Through these interrelated elements of the user interface, each independently developed program defines a distinctive path from the user's task to its solution. *E.g.*, *Lotus* at 67.

The user interface, including each of its constituent elements, could be expressed in many different forms. This is demonstrated by the substantially different user interfaces among computer programs performing similar generic tasks.

The design of each element of the user interface of a program is inextricably related to the design of the other elements. Their development is a creative task. Together these elements define the structure, sequence and organization of the user interface of the program. *Lotus* at 67, 63-64.

C. The Command Set or "Language" of the Program

The command set of a computer program consists of a compilation of declarative statements or expressions. This compilation of commands both reflects and expresses the

structure, sequence and organization of the user interface of the program. As is true with other aspects of the user interface, the compilation of commands varies substantially from one independently developed program to another, even among programs that perform the same general functions or tasks. *See, e.g., Lotus* at 63-64, 67.

The design of the command set is an integral part of the same creative process through which the program developer designs and expresses his or her distinctive approach to the solution of the generic tasks performed by a program. The selection of command sequences, and even individual commands, is closely related to the design of the sequence or flow of screens and the selection of individual screen content.

The *selection of which commands* to include in the program is often based upon the developer's creative judgment regarding the needs of potential users of the program and how best to structure, organize and express the desired communication between the program and user. The compilation of commands generally is intended to make the program versatile, efficient and relatively easy to use.

The *names or declarative statements chosen to express the commands* also represents original authorship. The compilation of phrases that comprise the command set may reflect the developer's efforts to select command names that are informative, easy to learn and remember, and intuitive to users or that conform to the developer's aesthetic sensibilities. The command set also has to meet certain design criteria imposed by other aspects of the user interface. For example, the decision to include menus of certain commands may require the developer to limit the number of these types of commands in order to avoid an unwieldy and unaesthetic menu. Thus, creative decisions made in authoring other aspects of the user interface may influence creative decisions in authoring commands, and vice versa.

The process of selecting commands, command names and syntax for a program is an intensely creative process, usually reserved for senior designers.

Although some command terms may be English-like,[4] the program explicitly defines the meaning of each command term, and it may respond to the command by performing in a manner quite different from what might be suggested by the English language meaning of the term. Each command has a special and narrowly defined meaning, determined by the specific actions the program will undertake in response to the command, by a specific structure and syntax for the command statement, by the input that the user must provide prior to execution of the command, by the output that will be presented to the user in response to the command, and by the specific sequences of commands and screens with which each command must be used.

[4] Other commands in a program invariably are not English-like.

Contrary to English language expressions, commands only make sense as defined and embodied in the particular program.[5]

Certain computer programs, such as the program in *Lotus*, allow the user to place in a file in a storage device in the computer a series of commands for later or repeated execution. Due to this feature of the program, the program's command set also sometimes is referred to as a "command language." *E.g.*, *Lotus* at 64 (the "macro language").

The command set or command language is not a "medium" like the English language. It is authored as an integral part of the user interface of the program, and itself represents a specific and highly structured dialogue or expression of commands tailored to the tasks that the program was designed to perform and tailored to the other parts of the program's user interface.

Not only do recent cases support the copyrightability of the command set of a computer program as an integral part of the user interface of the program, but 80 years of legal precedent firmly establishes the principle that original compilations of symbolic languages, declarative statements and codes are independently protectible under the copyright laws. Section III.C.2., *infra*.

II. THE LEGAL STANDARD IN AN ACTION FOR COPYRIGHT INFRINGEMENT

In order to prevail in a copyright infringement action, the plaintiff must establish two essential elements: (i) ownership of copyright and (ii) copying by the defendant. *Sid & Marty Krofft Television Productions, Inc. v. McDonald's Corp.*, 562 F.2d 1157, 1162 (9th Cir. 1977).

To prove ownership of a valid copyright, plaintiff must establish that the work constitutes copyrightable subject matter, that the work is original and that plaintiff has complied with applicable statutory formalities. *Lasercomb America, Inc. v. Holiday Steel Rule Die Corp.*, 656 F. Supp. 612, 614 (M.D.N.C.), appeal dismissed, 829 F.2d 36 (4th Cir. 1987). A copyright registration certificate is prima fade evidence of the validity of the plaintiff's copyright 17 U.S.C. § 410(c). The burden of proof is on defendants to overcome the presumption arising from the granting of a copyright certificate by the Copyright Office. *See Transgo, Inc. v. Ajac Transmissions Parts Corp.*, 768 F.241 1001, 1019 (9th Cir. 1985), *cert. denied*, 474 U.S. 1059, 106 S. Ct. 802 (1986).

[5] Some of the individual command names may be found in other computer programs. However, the meaning of even these commands generally will differ among programs, as will the sequences in which the commands may be used, the syntax of individual commands and, of course, the compilation of commands of which any individual command is only a small part. *E.g., see Lotus* at 67 (where the compilation of commands was distinct, despite some of the command terms being "quite obvious").

Copying by the defendant may be established in one of two ways: (i) by offering direct evidence of copying or (ii) by establishing that defendant had access to the copyrighted work and that defendant's work is substantially similar to the plaintiffs work. *Sid & Marty Krofft*, 562 F.2d at 1162.[6]

III. THE USER INTERFACE, AND EACH OF ITS CONSTITUENT ELEMENTS, CONSTITUTES COPYRIGHTABLE SUBJECT MATTER

Pursuant to the constitutional mandate "to promote the Progress of the Sciences and useful Arts," U.S. Coast. Art. I, § 8, C1. 8, in 1980 Congress amended the Copyright Act, 17 U.S.C. 101, 117, to make explicit that the Act protects computer programs.[7] Court decisions following the 1980 Amendment consistently have extended copyright protection to all aspects of computer programs, including the user interface or the "look and feel" to the user of the program. *Johnson Controls, Inc. v. Phoenix Control Systems, etc., et al.*, 886 F2.d 1173, 1175 (9th Cir. 1989) (copyright protection extends to structure and user interface of computer programs); *Telemarketing Resources v. Symantec Corp., Copyright L. Dec.* (CCH) ¶ 26,514 (ND. Cal. Sept. 6, 1989) ("Copyright protection applies to the user interface, or overall structure and organization of a computer program, including its audio-visual displays, or screen 'look and feel.'").

A. The Sequence, Structure and Organization of the User Interface is Copyrightable Subject Matter.

---

[6] Copyright infringement has frequently been determined as a matter of law where the works are so similar that the possibility of independent creation is precluded. *E.g., Twentieth Century-Fox Film Corp. v. MCA, Inc.*, 715 F.2d 1327, 1330 (9th Cir. 1983); *Lotus* at 68-70 (finding copying as a matter of law based upon the similarity of the user interface of two computer programs); *Lasercomb America, Inc. v. Holiday Steel Rule Die Corp.*, 656 F. Supp. at 616 (computer program); *Eisenman Chem. Co. v. NL Indus., Inc.*, 595 F. Supp. 141, 148 (D. Nev. 1984); *Silverman v. CBS, Inc.*, 632 F. Supp. 1344, 1352 (SD.N.Y. 1986), *judgment aff'd in part; vacated in part*, 870 F.2d 40 (2d Cir. 1989), *cert denied*, ___ U.S.___, 109 S. Ct. 3219 (1989) (script passages); *Knickerbocker Toy Co., Inc. v. Genie Toys, Inc.*, 491 F. Supp. 526, 529 (E.D. Mo. 1980); *Champion Map Corp. v. Twin Punting Co.*, 350 F. Supp. 1332, 1336 (E.D.N.C. 1971) (substantial similarities are sufficient to establish infringement in a summary judgment proceeding, even in the presence of a general denial of copying or the absence of proof of actual copying); *Peter Pan Fabrics, Inc. v. Dan River Mills, Inc.*, 295 F. Supp. 1366, 1369 (S.D.N.Y.), *aff'd*, 415 F.2d 1007 (2d Cir. 1969).

[7] The Computer Software Copyright Act of 1980, Pub. L. No. 96-517, Section 10, December 12, 1980, 94 Stat. 3028 ("1980 Amendment") embodied in 17 U.S.C. §§ 101, 117. In fact, it is generally accepted that computer programs were subject to copyright protection even before enactment of the 1980 Amendment. The legislative history of the 1976 Copyright Act makes it clear that Congress considered programs to be protected prior to 1980. *See, e.g.*, Notes of Committee on the Judiciary, H.R. No. 94-1476, *quoted at* 17 U.S.C.A. following § 102 ("computer programs . . . could be regarded as an extension of copyrightable subject matter Congress had already intended to protect, and were thus considered copyrightable from the outset without the need of new legislation").

The landmark decision in *Whelan Assoc., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1236 (3d Cir. 1986), *cert. denied*, 479 U.S. 1031, 107 S. Ct. 877 (1987), held that all aspects of a computer program not dictated by its overall function may constitute protectible expression. As the Third Circuit Court of Appeals explained:

> In other words, *the purpose or function of a utilitarian work would be the work's idea, and everything that is not necessary to that purpose or function would be part of the expression of the idea.*

Id. at 1236 (emphasis in original). *See also Dynamic Solutions, Inc. v. Planning & Control Inc., Copyright L. Dec.* (CCH) ¶ 26,062, p. 20,912 (1987) (relying on this test to find infringement by computer program that shared structural similarities to plaintiff's computer program).

Whelan has been adopted as the law by the Ninth Circuit and other Circuit Courts of Appeals. In *Johnson Controls, Inc. v. Phoenix Control Systems, Inc.*, 886 F.2d at 1175-76, the Ninth Circuit applied the test from *Whelan* and concluded that the structure, sequence and organization of a computer program and its user interface qualify for protection where there are alternative ways to design these aspects of the program. *See Telemarketing Resources, Copyright L. Dec.* (CCH) ¶ 26,514 at p. 23,085; *see also McCulloch v. Albert E. Price, Inc.*, 823 F.2d 316, 320 (9th Cir. 1987) (decorative plate intended to honor someone qualified for protection where other ways to express appreciation could have been used, such as plaques, gold watches or jeweled pins); *Sid & Marty Krofft*, 562 F. Supp. at 1171 (copyright infringement where defendants had many other ways to express the idea of a fantasy land with characters, but chose to copy plaintiffs expression of that idea).

To the same effect is *Broderbund Software, Inc. v. Unison Rog Inc.*, 648 F. Supp. 1127 (N.D. Cal. 1986). Relying on *Whelan*, the district court in Broderbund held that a computer program infringed plaintiff's copyright because it copied the structure and sequence of screens comprising the user interface of plaintiff's program:

> [T]he overall structure, sequence and arrangement of the screens, text, and artwork (*i.e.*, the audio-visual displays in general) are protected under the copyright laws.

Id., 648 F. Supp. at 1135.

The decision in *Broderbund* is a good illustration of how courts have distinguished between idea and expression in a computer program. The plaintiff in *Broderbund* sought protection for the user interface of a program that allowed users to print greeting cards, signs, banners and posters. Defendant's program copied the sequence of screens and certain content of individual screens from plaintiffs work. The defendant in *Broderbund* argued that the structure and sequence of screens constituted unprotectible ideas. Rejecting this argument, the district

court described the idea of the program as "the creation of greeting cards, banners, posters and signs that contain infinitely variable combinations of text, graphics and borders." *Id*. at 1133. By contrast, the district court held that the "structure, sequencing, and arrangement of screens" constitutes protectible expression because they could have been expressed in different ways.[8] *Id*. *See Manufacturers Technologies*, 706 F. Supp. at 994 (holding that copyright protection extends to the user interface, including the sequence or flow of screens and content of specific screens); *Digital Communications Assoc., Inc. v. Softklone Distrib. Corp.*, 659 F. Supp. 449, 459 (N.D. Ga. 1987) (finding infringement where defendant copied a single status screen of plaintiff's computer program); *Stern Electronics, Inc. v. Kaufman*, 669 F.2d 852, 857 (2d Cir. 1982) (sequence of audiovisual images in video game copyrightable).

B. The Lotus Case

Within recent months, the Massachusetts District Court published its landmark decision in *Lotus Development Corp. v. Paperback Software Int'l.* Judge Keeton's scholarly 115-page opinion (slip) held, as a matter of law, that the command driven user interface of Lotus 1-2-3 — including its "command structure" and the sequence and content of screens -- constitutes expression protected under the Copyright Act. Judge Keeton's opinion rejects virtually every argument against copyrightability and as a possible excuse to copying that is commonly advanced by the manufacturers of clones.

In *Lotus*, defendant Paperback Software International ("Paperback") copied aspects of the user interface of Lotus 1-2-3, a computer program sold by Lotus Development Corporation ("Lotus"). Lotus 1-2-3 is an integrated program including database management system support and a language" facility. *Lotus* at 43. The user interface of Lotus 1-2-3 is composed of sequences of screens, including individual screens listing available commands and organizing data for the user. Paperback advertised its product as a "workalike" of Lotus 1-2-3 (*id*. at 69), containing the same commands (*id*. at 69).

1. Copyrightable Subject Matter

The court in *Lotus* held that the structure, sequence and organization of the user interface of Lotus 1-2.3, including the screens and command set, constitute protectible expression. The district court's reasoning is compelling:

[8] The district court in *Broderbund* concluded that the structure and sequence of screens was expression by examining a third party's program that also allowed the printing of greeting cards and posters. Despite the similarity in the overall function of the programs, the sequence of screens in plaintiff's program differed from the sequence of screens in the third party's program. Based on this fact, the district court concluded there was room for individual expression in the structure, sequence and organization of the user interface, and the idea of a program to create greeting cards could be separated from the sequence or ordering of screens comprising the expression of the program's user interface. *Id*. at p. 1133.

This particular expression of a menu structure is not essential to the electronic spreadsheet idea, nor does it merge with the somewhat less abstract idea of a menu structure for an electronic spreadsheet. The idea of a menu structure -- *including the overall structure, the order of commands in each menu line, the choice of letters, words, or "symbolic tokens" to represent each command; the presentation of these symbolic tokens on the screen (i.e., fast letter only, abbreviation; full words, full words with one or more letters capitalized or underlined), the type of menu system used (i.e., one- two-, or three-line moving-cursor menus, pull-down menus, or command-driven interfaces), and the long prompts* -- could be expressed in a great many if not literally unlimited number of ways.

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. If particular characteristics not distinctive individually have been brought together in a way that makes the "whole" a distinctive expression of an idea -- one of many possible ways of expressing it -- then the "whole" may be copyrightable. The statutory provisions regarding "compilation," 17 U.S.C. §§ 101, 103, are not essential to this conclusion, but do reinforce it. A different total structure may be developed even from individual components that are quite similar and limited in number. To determine copyrightability, a court need not -- and, indeed, should not -- dissect every element of the allegedly protected work Rather, the court need only identify those elements that are copyrightable, and then determine whether those elements, considered as a whole, have been impermissibly copied. *Atari Games Corp. v. Oman*, 888 F.2d 878, 882-83 (D.C. Cir. 1989) (rejecting "component-by-component analysis," and ruling instead that focus must ultimately be on "work as a whole").

…

I conclude that a menu command structure is capable of being expressed in many if not an unlimited number of ways, and that *the command structure of 1-2-3 is an original and nonobvious way of expressing a command structure*. Emery Decl. ¶ 15. Accordingly, the menu structure, taken as a whole -- *including the choice of command terms, the structure and order of those teens, their presentation on the screen, and the long prompts* -- is an aspect of 1-2-3 that is not present in every expression of an electronic spreadsheet. It meets the requirements of the second element of the legal test for copyrightability.

Id. at 67-68 (emphasis added). Under *Lotus*, the structure, sequence and organization of the user interface of a computer program, including the command set and the sequence and content of screens, constitutes protected expression.

2. Copying

The district court in *Lotus* further held, as a matter of law, that Paperback copied the user interface of Lotus 1-2-3. *Id*. at 68. The district court found that Paperback's product contained the same command structure as Lotus 1-2-3 "-- that is, that defendants copied the expression embodied in the 1-2-3 menu hierarchy." *Id*. at 70.

3. Paperback's Unsuccessful Arguments to Excuse Copying

The district court in *Lotus* rejected virtually every argument against copyrightability and infringement advanced by clone manufacturers. First, Paperback argued that the user interface of Lotus 1-2-3 was a "useful article" or "functional" in nature, and thus not copyrightable. *Lotus*, as had a number of cases before it, soundly rejected this argument. *Id*. at 54, 71.

Next, Paperback argued that the command set of Lotus 1-2-3 is a "language" and "that languages are not copyrightable." *Id*. at 72. The district court in Lotus recognized that this argument is nothing more than a "striking word-game argument":

> Having explored the argument fully to try to understand its true nature, I conclude that defendants' "language" argument about the macro facility of Lotus 1-2-3, like defendants' "useful article" argument examined in Part IV(a), *infra, is totally without merit*.

Id. at 73 (emphasis added). *See* National Commission on New Technological Uses of Copyrighted Works ("CONTU"), *Final Report*, p. 53 ("copyright practice past and present . . . recognizes copyright protection for work of authorship regardless of the uses to which it may be put").

Finally, Paperback tried to excuse its copying. Paperback argued that duplication of the command structure of Lotus 1-2-3 was necessary to achieve "compatibility" or "standardization." Lotus 1-2-3 had become a *de facto* industry standard and "standardization," Paperback argued, was necessary to allow users to run programs developed with the commands of Lotus 1-2-3 on the other product. Other done manufacturers have made this same argument to excuse copying. The district court in *Lotus* firmly rejected this position, as have other cases in the computer industry in which it has been raised "the desire to achieve 'compatibility' or 'standardization' cannot override the rights of authors to a limited monopoly in the expression embodied in their intellectual 'work.'" *Lotus* at 69, 53, 71, 79. *E.g, Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F2d 1248, 1253 (3d Cir. 1983), *cert. dismissed*, 464 U.S. 1033, 104 S. Ct. 690 (1984).

Paperback also attempted to excuse its copying by stressing the differences between Lotus 1-2-3 and Paperback's product, as have other clone manufacturers. The district court easily disposed of this argument:

> Moreover, even if some elements of VP-Planner were very different, it would not give defendants a license to copy other substantial elements of 1-2-3 verbatim. If one publishes a 1,000-page book of which only a 10-page segment is an unauthorized reproduction of copyrighted material, and if the 10-page segment is a qualitatively substantial part of the copyrighted work, it is not a defense to a claim of infringement that the book is 99% different from the copyrighted material. [Citations omitted.] Thus, defendants' proof that VP-Planner has many features that are different from Lotus 1-2-3 is off point. The more relevant question is: does it have significant features that are substantially similar? I conclude, on the record before me, that there is no genuine dispute of material fact on this question. The answer to this question must be "yes."

Lotus at 70.

C. There Are Numerous Other Authorities that Establish that Each Element of the User Interface Constitutes Protectable Expression.

Each element of the user interface -- the sequence of screens, the content of individual screens, and the command set -- represents expression. Literally a legion of computer and non-computer cases supports copyright protection for the form of authorship embodied in each of these elements as well as the composite design of the user interface. Courts have consistently protected visual images, sequencing and order, compilations of words and phrases, and textual matter, regardless of the form of the work. The principles underlying these decisions apply with equal force to the user interface of a computer program.[9]

1. The Sequence of Screens Constitutes Protectible Expression.

It is beyond dispute that the generic tasks performed by most computer programs — their "idea" under *Whelan*, *Broderbund* and their progeny — could be implemented in different ways, including through a totally different sequence and flow of screens. In *Manufacturers Technologies, Inc. v. Cams, Inc.*, 706 F. Supp. 984 (D. Conn. 1989), the district court held a

[9] In granting broad copyright protection to computer programs, courts have relied on the broad protection against copying accorded to other forms of authorship, holding that computer programs should not be treated differently. *E.g.*, *Broderbund*, 648 F. Supp. at 1133; *Whelan*, 797 F.2d at 1234; *SAS Institute, Inc. v. S & H Computer Systems, Inc.*, 605 F. Supp. 816, 826 (D. Tenn. 1985); *Apple Computer, Inc. v. Formula Int'l, Inc.*, 562 F. Supp. 775, 783 (C.D. Cal. 1983).

sequence or flow of screens in a data base program for cost-estimating metalworking projects protectible against copying. The district court concluded that the sequence of screens was expression not dictated by the idea underlying the program:

> For all these reasons, the Court holds that the external sequencing and flow of plaintiffs screen displays in the creating-an-estimate sequence constitutes copyrightable expression which communicates to the user plaintiffs view of how a cost-estimate should be created.

Id. at 994. *Stillman v. Leo Burnett Co.*, 720 F. Supp. 1353, 1360 (N.D. Ill. 1989), also held that copyright protection extended to a short series of still screen shots in a television commercial, substantially fewer in number and containing substantially less authorship than the screens in the simplest of computer programs. *See Broderbund*, 648 F. Supp. at 1135, discussed at Section III.A., *supra*.[10]

The same principles of copyright law that require protection for the flow of screens in a computer program have been applied to virtually every other form of authorship. In this respect, the sequence of screens in a program is analogous to a compilation of forms and instructions in a daily organizer, the selection or organization of compiled facts, the detailed plot of a play, the format of a game show or television advertisement, and the outline of a book. In each of these contexts, the courts have long held that the structure, sequence and organization of a work of authorship constitute protectible expression. In *Harper House, Inc. v. Thomas Nelson, Inc.*, 889 F.2d 197, 204-05 (9th Cir. 1989), for example, the Ninth Circuit Court of Appeals held that "the selection, coordination and arrangement of elements in a daily organizer is copyrightable" relying on cases holding that the format of a country music show constitutes protectible expression.[11]

---

[10] *See also Whelan*, 797 F.2d at 1240 ("Congress intended sequencing to be protectible"); *Williams v. Arndt*, 626 F. Supp. 571, 578-79 (D. Mass. 1985) (holding that computer program infringed copyright in a manual for trading commodities since it copied "a detailed, step-by-step procedure or process to accomplish a specific desired result"); CONTU Report, pp. 53-54, 53 n.126 ("[f]low charts . . . are works of authorship in which copyright subsists"; a "flowchart is a graphic representation for the definition, analysis or solution of a problem in which symbols are used to represent operations, data flow, or equipment").

[11] "*See C. Blare & D. Richman, Inc. v. 20/20 Advertising*, 674 F. Supp. 671, 677-78 (D. Minn. 1987) (infringement of television advertisement because of similar sequencing, despite lack of any duplication of dialogue); *Bradbury v. Columbia Broadcasting System*, 287 F2d 478, 485 (9th Cir.), *cert. denied*, 368 U.S. 801, 82 S. Ct. 19 (1961) (infringement of books by television production based upon copying of sequence of events, incidents, development and interplay of characters); *Grove Press, Inc. v. Greenleaf Publishing Co.*, 247 F. Supp. 518, 525 (E.D.N.Y. 1965) (copyright protection for novel includes the pattern employed and the plot, plan, arrangement and combination of materials); *Meredith Corp. v. Harper & Row, Publishers, Inc.*, 378 F. Supp. 686, 690 (S.D.N.Y.), *aff'd*, 500 F.2d 1221 (2d Cir. 1974) (preliminary injunction against the sale of a textbook where defendant copied "the entire structure and topical sequence," despite absence of copying of text).

Those recent cases applying copyright protection to the structure, sequence and organization of the user interface of a computer program have simply embraced long-standing principles of copyright law. Under these principles, the sequence and flow of screens comprising the user interface of a computer program constitute protectible expression.

2. The Compilation of Commands Constitutes Protectible Expression.

a. The Command Set is Protectible as an Integral Part of the Design of the User Interface.

The efforts of the developers of the command structure of a computer program are the same as the efforts of other program designers that the courts have traditionally protected:

> [Throughout the preparation of a complicated computer program . . . the author is faced with a *virtually endless series of decisions* as to how to carry out the assigned task. Beginning with a broad and general statement of the overall purpose of the program, the author must decide how to break the assigned task into smaller tasks, each of which must in turn be broken down into successively smaller and more detailed tasks. At the lowest levels the detailed tasks are then programmed in source code. *At every level, the process is characterized by choice,* often made arbitrarily, and only occasionally dictated by necessity.

SAS Institute, Inc., 605 F. Supp. at 825 (finding infringement based on copying overall organization of the computer program) (emphasis added).

The command set of a program represents just such a design composite, resulting from literally thousands of decisions regarding what commands to include, what words or phrases should be used to identify them, in what sequences they must be used, what input should precede each command, in what syntax command statements must be made, what screens should appear when the command is selected, what prompts on the screens should assist the user to identify the various options available to the user, what command options should be given to the user at any point in the program's operation, etc. *See Lotus*.

The design of the command set is an integral part of the design of the user interface, including the sequence or flow of screens and the content of individual screens. The command structure largely defines the sequence, structure and organization of the user interface. Thus, the designers of the command structure of a program engage in the purest form of authorship. Their original creations are protectible under the copyright laws. Section II.C.1., *supra*, and authorities cited therein.

b. The Command Set is Protectible as a Compilation.

Although not necessary to copyrightability, the command set of a program is protectible as a compilation under the copyright laws. *Lotus, supra*, at 67. The Copyright Act specifically identifies original compilations as a form of authorship protected under the copyright laws. 17 U.S.C. § 103. A compilation is "a work formed by the collection and assembling of preexisting materials or of data that are selected, coordinated or arranged in such a way that the resulting work as a whole constitutes an original work of authorship." 17 U.S.C. § 101.

Where words, data or other materials are selected, coordinated and arranged in an original manner, the resulting compilation is protectible, even if the individual elements comprising the compilation are unprotectible ideas or otherwise in the public domain. Protection extends to the "selection, design and synthesis" in the compilation, not resulting novelty.[12] *United States v. Hamilton,* 583 F.2d 448, 451-52 (9th Cir. 1978) (elements of compilation amounting to "more than a matter of trivial selection" and originality in "taking the commonplace and making it into a new combination or arrangement" are protectible); *Financial Information, Inc. v. Moody's Investors, Service, Inc.*, 751 F2d 501, 507 (2d Cir. 1985) (only "a modicum of selection, coordination, or arrangement" is necessary to achieve protection).

The Ninth Circuit Court of Appeals explained these principles while rejecting a contention that a map lacked sufficient originality because it was merely a synthesis of information already depicted on maps in the public domain:

> Originality requires only that the work display "something irreducible, which is one man's alone," not that the work be novel in comparison with the works of others. When a work displays a significant element of compilation, that element is protectible even though the individual components of the work may not be, for originality may be found in taking the commonplace and making it into a new combination or arrangement.

Hamilton, 583 F.2d at 451 (citations omitted).

More recently, in *Feist Publications, Inc. v Rural Telephone Service Co., Inc.*, ___ S. Ct.____ (March 27, 1991), the Supreme Court affirmed that a minimal degree of creativity in selection or arrangement is all that is required to make a compilation protectible. Although the Supreme Court in *Feist Publications* concluded on the facts before it that neither

---

[12] As noted in the House Report supporting 17 U.S.C. § 103:

> A "compilation" results from a process of selecting, bringing together, organizing, and arranging previously existing material of all kinds, regardless of whether the individual items in the material have been or ever could have been subject to copyright.

Notes of Committee on the Judiciary, H.R. No. 94-1476, *quoted* at 17 U.S.C.A. following § 103.

the selection nor the arrangement in alphabetical order of names and addresses in a telephone directory was sufficiently original to warrant copyright protection, the Court emphasized that "the originality requirement is not particularly stringent" and that "the vast majority of compilations will pass this test." The Court's description of the creative process of original authorship that would render a compilation of facts protectible applies equally well to the creative process of creating a compilation of commands:

> The compilation author typically chooses which facts to include, in what order to place them, and how to arrange the collected data so that they may be used effectively by readers. These choices as to selection and arrangement, so long as they are made independently by the compiler and entail a minimal degree of creativity, are sufficiently original that Congress may protect such compilations through the copyright laws.

The author of a compilation of commands chooses which commands to include, what words or phrases should be used to identify those commands, the sequence or order in which the commands appear in the program, and how to arrange and present the commands on screen so that they may be used effectively. Under *Feist Publications*, it is clear that only a minimal degree of creativity and originality is required to make such a compilation protectible.

The user interface of computer programs has been accorded protection as a compilation in several cases. For example, the compilation of parameter/command terms on a single status screen of a computer program was held to constitute copyrightable subject matter in *Digital Communications Assoc., Inc.*, 659 F. Supp. at 463.[13] *Lotus* at 80 (selection of commands and screens); *Manufacturers Technologies, Inc. v. Cams, Inc.* (selection of screens in data base program).

The same principles that led the courts in these and other cases to find elements of the user interface protectible as compilations have been applied for more than 70 years in numerous cases that provide direct precedent for the protection of the command set of a computer program as a compilation under the copyright laws. CONTU's Final Report to Congress on copyright protection for computer programs, for example, quoted from Judge Learned Hand's seminal opinion in *Reiss v. Nat'l Quotation Bureau*, 276 F. 717 (S.D.N.Y. 1921), also cited approvingly in *Lotus* at 49. That case held that a compilation of coined words of five letters each, to be used as a code for transmitting telegrams and cables, was protectible expression. Judge Hand concluded that such a compilation of words represented the 'writing of an author." So too must a

[13] The district court did not directly address whether the command terms themselves were protectible, but did note that "the defendants could have used a wide variety of techniques to indicate which symbols the user should type to effectuate a command, *e.g.*, different symbols could have been chosen . . . ." *Id.* at 460.

compilation of commands -- or special code -- for communications between a user and a computer program.[14]

Professor Nimmer had occasion to comment generally on the protectibility of a compilation of words, and specifically on Judge Hand's opinion in *Reiss v. Nat'l Quotation Bureau*, in describing CONTIJ's Final Report to Congress:

> 15. Thus, CONTU clearly recognized that the protection of a program extends to the words or short phrases (whether mnemonics or other terms) as they are included in the expression of a program. We benefitted in this respect from Judge Learned Hand's opinion in what I have called in my *Treatise* the "touchstone" for interpreting the constitutional requirement that a copyrightable work must be a "writing," *Reiss v. National Quotation Bureau*, 276 Fed. 717, 719 (S.D.N.Y. 1921), quoted at CONTU Report, p. 14. Judge Hand there found copyrightable a series of meaningless words coined for use as a code for transmitting telegrams and cables.[15]

Based upon these principles, Justice A. Hand, in *Hartfield v. Peterson*, 91 F.2d 998 (2d Cir. 1937), held that cable and telegraphic codes constitute copyrightable subject matter. Although Justice Hand acknowledged that compilers of such code phrases "collect them from existing sources" and "alphabetically arrange them under appropriate headings," he held that this "would afford no justification . . . for copying an arrangement or sequence of [such] phrases which was the work of plaintiff. *Id*. at 999. Justice Hand explained his holding in the following terms:

---

[14] Judge Hand went on to suggest that even a new form of "abstract speech, with inflections, but as yet with no meaning, a kind of blank Esperanto," could be protectible under the copyright laws. *Id*. at 718.

[15] Professor Nimmer explained:

> 14. Referring to . . . the protectability of mnemonics or words used in a program, it is important to distinguish between the words standing alone, on the one hand, and the compilation and the usage of a compilation of words or mnemonics in context, on the other. CONTU was aware that, under traditional copyright principles, individual words alone and short phrases alone have been viewed as lacking sufficient copyrightable subject matter. We were also aware, however, of the equally traditional principle that it would be a *reductio ad absurdum* to deny protection to a program because the first word — or statement, or instruction — individually is uncopyrightable, as is the second individually, etc. Were it otherwise, the copyright on *Gone With the Wind* would likewise be invalid because its first word is not copyrightable, its second is not, and so forth.

> It must be borne in mind that the compilation is the sum total of the words and phrases as arranged by the author and that the copyright is valid because of the originality of the combination.

Id. at 1000.

Other cases have accorded protection to compilations of words and short statements in analogous contexts. *See College Entrance Book Co. v. Amsco Book Co.*, 119 F.2d 874 (2d Cir. 1941) (holding that a list of commonly used French words compiled for use in an exam preparation booklet was protectible against copying, even though defendant's list included only 15% of the words in question); *Applied Innovations, Inc. v. Regents of the Univ. of Minn.*, 876 F.2d 626, 634 (8th Cir. 1989) (infringing computer program where defendant copied certain short, simple statements such as "I am a good mixer" and "[n]o one seems to understand me").[16]

The command set of a computer program plainly represents authorship with at least as much claim to protection as the lists of words and phrases in these cases.

c. The Command Set is Protectible Against a Competitor Who Would Secure an Unfair Advantage by Copying.

Another line of cases in the computer industry supports protection from copying for the compilation of commands in a computer program. Under these cases, aspects of the interface of a protected work that would allow another to benefit materially from the efforts of the copyright holder are protected from copying for use in a competitive product. In *West Pub. Co. v. Mead Data Cent., Inc.*, 799 F.2d 1219 (8th Cir. 1986), *cert denied*, 479 U.S. 1070, 107 S. Ct. 962 (1987), defendant Mead proposed to include in LEXIS a reference system that would allow users to determine upon what page in the West Reporter System particular text of a case could be located. The Court of Appeals held that defendant's pagination system was infringing and affirmed the entry of a preliminary injunction. The Court of Appeals found that West's arrangement of cases in its reporter system

[16] To the same effect is *Trebonik v. Grossman Musk Corp.*, 305 F. Supp. 339, 348 (ND. Ohio 1969), where the district court found protectible an arrangement of guitar chords and the nomenclature used to describe the chords. Despite the fact that the individual chords clearly were not protectible, the district court concluded that this particular compilation was different than other compilations of chords, and that there were a large number of descriptive terms that could have been used to denote any given chord. *See Raffoler Ltd. v. Peabody & Wright, Ltd.*, 671 F. Supp. 947, 951 (E.D.N.Y. 1987) (combination of short phrases, protectible as an original listing); *Marling v. Ellison, Copyright L. Dec.* (CCH) ¶ 25,586, p. 18,460 (1982) (lists of foreign language terms categorizing menu items, followed by translations, held protectible); *see also Pantone, Inc. v. A.I. Friedman, Inc.*, 294 F. Supp. 545, 548 (S.D.N.Y. 1968) (holding protectible a selection of colors, blends of colors, and presentation arrangement).

> is a copyrightable aspect of its compilation of cases, that the *pagination of West's volumes reflects and expresses West's arrangement*, and that MDCs intended use of West's page numbers infringes West's copyright in the arrangement.

Id. at 1223 (emphasis added).

The Court of Appeals specifically stated that this result was appropriate whether or not the page numbers themselves constituted protectible expression:

> [P]rotection for the numbers is not sought for their own sake. It is sought, rather, because access to these particular numbers . . . would give users of LEXIS a large part of what West has spent so much labor and industry in compiling, and would *pro tanto* reduce anyone's need to buy West books.

Id. at 1227. *See also Worlds of Wonder, Inc. v. Venial Learning Systems, Inc.*, 658 F. Supp. 351, 355 (N.D. Tex. 1986) (holding a competitor's cassette tapes that could be used with the Teddy Ruxpin animated toy bear infringing on the grounds that the tapes produced substantially similar results when played in the command track for the toy bears sold by plaintiff); *Worlds of Wonder, Inc. v. Vector Intercontinental, Inc.*, 653 F. Supp. 135, 139 (N.D. Ohio 1986) (same); *Pearl Systems, Inc. v. Competition Electronics, Inc., Copyright L. Dec.* (CCH) ¶ 26,338 (1988) (copying of two features of interface of a shot timing device held to constitute infringement where those two features played an important role in the marketplace because of their frequency of use).

Analogous to West's pagination system, or the command track for Teddy Ruxpin in the *Worlds of Wonder* case, the command set of a program both reflects and expresses the structure, sequence and organization of the user interface. Section I.B.3., *supra*. Its unauthorized use by a competitor would allow the competitor to secure an unfair advantage from the efforts of the program developer. Moreover, unlike the West pagination system, the command set of a program itself is expressive and requires original and creative effort to compile and design.

Accordingly, the command set of a computer program is protectible under the copyright laws (i) as an integral part of the user interface of the program; (ii) independently as an original compilation of commands; and (iii) because it reflects the selection, sequence and content of the user interface of the program, the benefits of which a clone manufacturer may gain unfairly by copying.

3. The Organization, Layout and Content of the Individual Screens Constitute Protectible Expression.

The composition of the screens, including their menu listings, distinctive layout and presentation of information may be expressive and original. Like the status screen in *Manufacturers Technologies*, 706 F. Supp. 984 (D. Conn. 1989),[17] status displays may represent a particular means of apprising the user of the status of selected operations of the program, including its current operation, the name of the active file, its current drive, the number of options in a menu, which menu is highlighted, the record at which the user is positioned, and whether certain keys have been pressed. The design of this aspect of the user interface requires a decision as to what information should be included from the universe of information relating to the program operation, and it may include a particular arrangement of the information followed by related original text.

Similarly, the selection, organization, layout and text of the menu listings and commands, reflected on screens, may be original and expressive. Section II.B., *supra*.

IV. COPYRIGHT PROTECTION AGAINST COPYING BY CLONES

Typically, clone manufacturers have admitted, or necessarily must admit, that the user interface, including the command structure, of their products was copied from the original program developer. Often the clone manufacturer has used the similarity of the user interface as the centerpiece of his or her marketing plan.

Based on such admissions of actual copying, a court may not need to address the issues of access and substantial similarity. Section III., *supra*. It may be clear from the admissions that the two-part test for determining substantial similarity has been met. First, the so-called "extrinsic" test is applied to determine whether there is a similarity in the ideas underlying the two works. Second, the "intrinsic" test is applied to determine whether there is substantial similarity in the expression of the two works, based upon the response to the works by the ordinary person. *Sid and Marty Krofft,* 562 F.2d at 1164.

Clone manufacturers consistently argue, however, that they made changes or improvements" to the program. Such arguments should be irrelevant as a matter of law. Infringement is determined by similarities in the product, not by whether defendant has made

---

[17] In that case, the district court explained:

> The idea of apprising the user of the status of one's efforts in cost-estimating a part is not copyrightable. Plaintiff's expression is not a necessary incident to this idea. *That expression reflects selection as to what should be made part of the status report arrangement of the terms there, assignment of numbers to specific operations/ departments and tools, and coordination in the manner of building on the status report as the user progresses through various steps.* Therefore, it is copyrightable and subject to protection if infringed.

Id. at 996 (emphasis added).

some additions or enhancements to the copy. *E.g.*, *Lotus* at 69-70; *Sid & Marty Krofft*, 562 F.2d at 1165-66; *Illinois Bell Tel. Co. v. Haines & Co., Inc.*, 683 F. Supp. 1204, 1209-10 (N.D. Ill. 1988), *aff'd*, 905 F.2d 1081 (7th Cir. 1990) (changing the order in which elements are presented, editing, adding new data, and Changing some listings to boldface type, do not avoid a claim for infringement of a compilation); *United Tel. Co. of Mo. v. Johnson Publishing Co.*, 671 F. Supp. 1514, 1521-22 (W.C. Mo. 1987), *aff'd*, 855 F.2d 604 (8th Cis. 1988) (summary judgment of infringement, holding that adding original material to or changing a copied compilation does not excuse copying); *Whelan*, 797 F. Supp. at 1246; *SAS Institute*, 605 F. Supp. at 829-30 ("the piracy of even a quantitatively small fragment [of a computer program] may be qualitatively substantial"); *Marling v. Ellison, Copyright L. Dec*. (CCH) ¶ 25,586 at p. 18,460 (replication of 1100 out of 1800 entries or 1325 of 1925 entries sufficient for finding of infringement); *Baxter*, 812 F.2d at 425 (copying of a qualitatively important part of a work is sufficient, even if it is a small part of the whole).[18]

Closing Argument

Copyright law provides protection for "original works of authorship." Traditionally, "original works of authorship" have included such articles as novels, poetry, painting, sculpture and music — the creative achievements of gifted individuals. More recently, Congress and the courts have concluded that computer programs also are original works of authorship and that the originality and creativity that goes into the development of computer programs is just as deserving of copyright protection as any other form of original authorship.

The copyright laws require that original authorship be protected in whatever form it may take. The command structure of a computer program is original authorship and is entitled to the same protection that is accorded to other works of authorship and other aspects of a computer program. The creativity and imagination that goes into the development of an elegant, well-designed command structure is the same type of original effort that goes into other aspects of computer program development. Especially as the nature of software development and computer technology itself are changing today, it would be shortsighted to allow free copying of the external design of computer programs and effectively limit competition and innovation to writing better code. The structure, sequence and organization of the user interface increasingly are central to software's innovative quality and value.

[18] Contentions that a program is different because it is "faster" should be similarly irrelevant. *Midway Mfg. Co. v. Arctic Int'l Inc.*, 704 F.2d 1009, 1013 (7th Cir.), *cert. denied*, 464 U.S. 823, 104 S. Ct. 90 (1983) (modification of computer program for video game in a manner that causes operation of program to "speed-up" constitutes infringing creation of derivative work). So is any argument that the clone manufacturer has invested its own efforts to achieve the alleged performance enhancements. *Lasercomb America*, 656 F. Supp. at 616 (entering summary judgment of liability for copyright infringement of computer program, despite evidence that defendant invested substantial labor and expense to develop its competing program).

B. Some Issues Are Back: Oracle v. Google (Fed. Cir. 2014)

In *Oracle v. Google*, the Federal Circuit held that the application programming interfaces, including the command language and syntax, and organization, structure and sequence of the APIs, may be appropriate subject matter for copyright protection. This decision may represent a substantial shift in copyright law as applied to software, including in elements relevant to interoperability, compatibility and industry standards.

Prior to the *Oracle* decision, the case that most closely considered these issues was *Lotus v. Borland*, 49 F.3d 807 (1st Cir. 1995), *aff'd without opinion, by an equally divided Court*, 516 U.S. 233 (1996). In *Lotus*, the First Circuit held that command structures are not protectable under the copyright law, as representing a method of operation. Since the Supreme Court decision in *Lotus v. Borland* was by an equally divided Court, it served to affirm the decision of the First Circuit and only stands as binding precedent in the First Circuit.

For the past 19 years, however, many have believed that the decision in *Lotus v. Borland* would guide the law of copyright as applied to computer software, including command structures and similar elements of software. But the issue remained undecided.

This no longer can be assumed to be the case (if it ever could). The Federal Circuit expressly rejected *Lotus v. Borland* in the *Oracle* decision.

The Federal Circuit's decision in *Oracle v. Google* was based on its interpretation of Ninth Circuit law. The Court remanded for consideration of Google's fair use defense, consistent with the guidance on that issue set forth in the Court's opinion.

Copyright Law as Applied to Computer Programs

"Copyright protection subsists … in original works of authorship, fixed in any tangible medium of expression, now known or later developed, from which they can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device." (Copyright Act, Sec. 102(a))

The list of categories of protectable works includes "literary works" -- under which computer programs are protectable: "'Literary works' are works … expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as books, periodicals, manuscripts, phonorecords, film, tapes, disks, or cards, in which they are embodied." Sec. 101.

Computer programs are defined as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." Sec. 101.

Some cases have held that the scope of protection for software may include, among other things: source and object code; elements of the user interface; the logic and engineering of the program (at some level); program sequence; and sequences of screens.

Outside the First Circuit, where *Lotus v. Borland* was decided, authority on the protection of command structures and computer languages generally has not been clear.

Important for the protection of information technologies, compilations may be copyrightable subject matter. "A 'compilation' is a work formed by the collection and assembling of preexisting materials or of data that are selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship." A protectable compilation may consist of facts or public domain data -- or as the Court reasoned in *Oracle v. Google* (*infra*), a command language. Protection for a compilation "extends only to the material contributed by the author of such work, as distinguished from the preexisting material." Thus, it is the selection, organization and/or arrangement that may be protected.

Examples of other subject matter that has been discussed in individual cases as comprising or potentially comprising copyrightable subject matter include: flow charts; new form of abstract speech, including languages; legal pagination scheme; shot timing; animated toy bear; selection of guitar chords and code to represent them; selection of colors and presentation arrangement; games and images with characters; and the sequencing of images in a television commercial (see the article comprising section IA, *supra*).

The Federal Circuit Decision

"Because we conclude that the declaring code and the structure, sequence and organization of the API packages are entitled to copyright protection," the Federal Circuit reversed the trial court's decision.

Oracle's predecessor, Sun Microsystems, wrote a number of ready-to-use Java programs to perform common computer functions, organizing them into "packages," which were the APIs at issue in the case. As explained by the Federal Circuit, every API consists of (1) the "declaring code" or "header," which is the expression that identifies the prewritten function, and (2) implementing code for the API. The header specifies the inputs, name and functionality of the API. The programmer uses the declaring code as commands to the computer to execute the associated implementing code for the "declared function."

Google created 168 APIs for the Android platform. Google copied from Oracle's program the declaring code for 37 of the APIs. "Google believed that Java application programmers would want to find the same 37 sets of functionalities in the new Android system callable by the same names as used in Java." *Id.* at 10. According to the Federal Circuit:

> In doing so, Google copied the elaborately organized taxonomy of all the names of methods, classes, interfaces and packages -- the "overall system of organized names -- covering 37 packages, with over six hundred classes, with over six thousand methods." *Id.* at 11.

The implementing code was independently written by Google (with limited exceptions).

The Federal Circuit held that these copied elements of the APIs were copyrightable. Specific holdings of the Court likewise will be important to other cases and analyses.

The Court specifically held that the trial court "erred by importing fair use principles, including interoperability concerns, into its copyrightability analysis." *Id.* at 17. This and similar reasoning (*infra*) will be important in determining the scope of protection in contexts of compatibility and interoperability, licensing analyses, and defenses to infringement.

The Federal Circuit specifically held that the analysis of the application of the merger doctrine should be made at the time of the creation of the copyrighted work, not at the time of copying. Thus, it may not be a defense to infringement that copying of certain elements was necessary to compatibility, at least to the extent these elements were originally created by the copyright owner.

Similarly, the Federal Circuit stated the application of the *scenes a faire* doctrine must be based on usage at the time of the creation of the copyrighted work, not at the time of copying. The doctrine must focus "on the circumstances presented to the creator, not the copier." *Id.* at 37. Thus, the fact that elements of the copyright owner's program have come into common usage, *e.g.*, for compatibility, may not mean that these elements can be copied.

The Federal Circuit explained that the fact that a program becomes successful, and elements of it become industry standards, does not necessarily inject those elements into the public domain or remove the protection of the copyright laws from the program developer.

The Federal Circuit specifically held that compilations of words or phrases may be protectable, even if copyright law might not protect short phrases or words alone.

Critically, the Federal Circuit rejected the "method of operation" analysis/conclusion of *Lotus v. Borland*. The Federal Circuit observed that computer programs are, by their nature, functional works. Nonetheless, Congress determined that they are protectable under the copyright laws.

The Federal Circuit "conclude[d] that a set of commands to instruct a computer to carry out desired operations may contain expression that is eligible for copyright protection…. [A]n original work--even one that serves a function--is entitled to copyright protection as long as the author had multiple ways to express the underlying idea." *Id.* at 42-3.

II. Open Issues -- Will Lightening Strike Twice?

Of course, the issues raised in the Oracle/Google litigation are not finally resolved. In addition, many other questions regarding the protectability of the non-literal elements of software remain open to debate. Might lighting strike twice? Will there be further development of the law so that software developers, in Professor Nimmer's words, have "adequate guidelines regarding what level of independent development is required to avoid copyright infringement." (*Supra*.) The following identifies selected issues regarding the scope of ownership rights in software as to which important questions remain. Some issues remain open in the sense that the

answers given vary from circuit to circuit. With respect to other issues, they simply have not been clearly or adequately addressed. Language or holdings in different cases can be found to support either side of virtually every one of the following issues.

At what level of abstraction is the internal structure, organization or engineering protectable?

- Functional composite, detailed tasks, organization of modules, internal program interfaces
- Program behavior
- Program sequencing
- Idea/procedure/system v. expressive design elements/compilation?

What elements of the external structure, organization and engineering may be protectable?

- Programming language
- Commands, syntax and/or mnemonics
- Access protocols
- Sequencing in user interface (e.g., screen sequence)
- Behavior of program (e.g., inputs, outputs, user functions)
- Method of operation/procedure/system v. expressive design/communication/information/flow?

Is there a test for idea/method/system v. expression in a utilitarian work such as a program?

- Choice not to copy -- are there alternative ways to do (express) the same thing
- Compatibility considerations
  - -- clones
  - -- add-on products
  - -- interoperability
  - -- user education
- Merger
  - -- balance expression v. functional aspects
  - -- number of alternatives
  - -- where both method of operation and expression

Abstraction-filtration-comparison?

- Is it a test for scope of protection v. a logical approach to organization of a case
  - -- The hard questions
  - -- "Describing this approach as abstraction-filtration-comparison should not convey a deceptive air of certitude about the outcome." *Engineering Dynamics, supra.*
- What of the above do you filter out
- What are "external constraints" -- elements authored by third parties, authored by the copyright owner
- Treatment of compilations when filter individual elements

Test for infringement?

- How close a match -- e.g., substantial similarity v. virtual identity v. material element
- How to treat a compilation following filtration of individual elements

What may be a fair use?

- Reverse engineering/decompilation
  -- game cases: no contractual restrictions versus
  -- software: contractual restrictions to preserve secrecy of source code
  -- does fair use preempt contract and/or trade secret law
- Is subject to widespread or standard usage enough
- Role of interoperability/compatibility
- Role of licensing for extended commercial use

Scope of waiver, estoppel and implied license defenses