2013-1021, -1022

IN THE

United States Court of Appeals

FOR THE FEDERAL CIRCUIT

ORACLE AMERICA, INC.,

Plaintiff-Appellant,

—v.—

GOOGLE INC.,

Defendant-Cross Appellant.

ON APPEAL FROM THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF CALIFORNIA
JUDGE WILLIAM H. ALSUP
CASE NO. 10-CV-3561

BRIEF FOR *AMICUS CURIAE* RALPH OMAN SUPPORTING THE POSITION OF PLAINTIFF-APPELLANT AND URGING REVERSAL

MARCIA B. PAUL
LACY H. KOONCE, III
DEBORAH A. ADLER
DAVIS WRIGHT TREMAINE LLP
1633 Broadway
New York, New York 10019
(212) 489-8230

*Attorneys for Amicus Curiae
Ralph Oman, former Register of
Copyrights of the United States*

CERTIFICATE OF INTEREST

Counsel for amicus curiae Ralph Oman certifies the following:

1. We represent Ralph Oman.

2. That is the real name of the real party in interest.

3. Ralph Oman is an individual.

4. The following law firm and partners or associate are expected to appear for Ralph Oman in this Court:

Davis Wright Tremaine LLP

Marcia B. Paul
Lacy H. Koonce, III
Deborah A. Adler

Date: February 19, 2013 By: /s/ Marcia B. Paul
Attorney for Amicus Curiae Ralph Oman

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

RULES

OTHER AUTHORITIES

I.

INTRODUCTION AND STATEMENT OF INTEREST

Amicus curiae Ralph Oman was the Register of Copyrights of the United States (1985 to 1993). As Register, he acted as principal advisor to Congress on copyright policy and testified more than forty times on proposed copyright legislation and treaties, and on the state of the Copyright Office. From 1975 to 1977, he served as Chief Minority Counsel on the Subcommittee on Patents, Trademarks and Copyrights, and from 1982 to 1985 as Chief Counsel of that reconstituted subcommittee. He was personally involved in the final stages of the drafting and passage of the Copyright Act of 1976, 17 U.S.C. §§ 101, et seq. (the "Copyright Act"). During Mr. Oman's tenure as Register, he was responsible for helping transition United States copyright law from the analog to the digital age, and was part of the government team that convinced the world community to protect computer software as a literary work under national copyright laws and treaties.

Mr. Oman is now the Pravel Professorial Lecturer in Intellectual Property and Patent Law at the George Washington University Law School, where he has taught copyright law for twenty years. He is a former trustee of the Copyright Society of the United States, former chair of the ABA's Copyright Division, and a member of the governing council of the ABA's Intellectual Property Law Section.

In 2002, he received the Jefferson Medal in recognition of his lifelong commitment to intellectual property. Mr. Oman cares deeply about the continued vitality of the copyright system and proper interpretation of the law. For that reason, he believes it imperative that this Court correct the misreading of the Copyright Act which permeates the opinion of the court below regarding the copyrightability of Oracle's software (the "Decision"),[1] specifically the court's restrictive analysis of the availability of copyright protection for non-literal aspects of that software. While Oracle argues in the first instance that Google copied extensive verbatim portions of the Java source code (the "expressive declaring code", Br. at 40), and thus is liable for copyright infringement since it has no valid fair use defense,[2] the focus of this brief is upon the denial of copyright protection to the creative structure, sequence and organization of Oracle's software packages. That aspect of the court's decision, if not reversed, will chill investment and innovation in the software industry, and retard development of future generations of software,

[1] Pursuant to Federal Rule of Appellate Procedure 29(c)(5), counsel and Mr. Oman are the sole authors of this brief; no counsel for a party authored this brief in whole or in part. No party or party's counsel contributed money that was intended to fund preparing or submitting this brief, and no person other than the amicus curiae contributed money intended to fund preparation or submission of this brief. Pursuant to Federal Rule of Appellate Procedure 29(a), all parties have consented to the filing of this brief.

[2] Mr. Oman takes no position on the application of the law to the specific factual record in this case, including the viability of Google's fair use defense or with respect to the patent claims at issue.

because it largely eviscerates copyright protection for some of the most creative aspects of computer software.

The court's result appears to have been outcome-driven, due presumably to the popularity of the Java platform and a belief that patents should be the only or primary means of protecting software. The result and the underlying reasoning are inconsistent with legislative intent and Copyright Office policy with respect to the key issue in this case: the extent to which copyright protects expression in computer programs.

Congress has spoken clearly and unequivocally on the subject: expressive elements of software are protected by copyright, and software must be analyzed for protectability consistent with traditional copyright principles (which recognize expressive content in both literal and non-literal authorship). This includes protection for original structure, sequence, and organization of software. The full copyright protection Congress accorded to software extends to interactive features that, despite their functional aspects, must be individually and collectively examined to determine whether or not they represent expressive protectable elements.

The court below misconstrued the fundamental idea/expression dichotomy and its corollary that expressive aspects of functional works are protectable as long as the expression does not merge with the function. In so doing, the court created an unwarranted *de facto* and *sui generis* exception to the broad scope of copyright protection that Congress, the courts, and the Copyright Office have long accorded software. The court's reasoning reflects a mistaken view that non-literal components of software – especially ones that impact "interoperability" – are functional for all purposes and cannot qualify for copyright protection. Under the trial court's reasoning, even ***expressive*** software elements are not protectable if they bear on interoperability. That is not the law as enacted or as previously interpreted, and it should not be the law as a matter of policy. Copyright law protects expressive and original structure, sequence and organization, even if it functions as part of a "command structure" of computer software.

The court's concern that given the popularity of Oracle's Java platform, third parties should be allowed to create compatible products, led it astray, causing it to issue a sweeping holding that threatens to swallow the existing statutory regime. That regime strikes a careful balance between protection of original authorship in software on the one hand, and the rights of third parties to utilize mere ideas and functional aspects of that software, on the other. To the extent protected expressive content must be copied to understand functionality, courts rely on the fair use defense to balance an author's rights against the public interest; that is the proper test for Google's actions here. By injecting the fair use balance into the threshold determination of copyright protection, the court below erred.

II.

THE TRIAL COURT ERRONEOUSLY FOUND THAT EVEN ORIGINAL STRUCTURE, SEQUENCE AND ORGANIZATION OF COMPUTER SOFTWARE IS UNPROTECTABLE IF IT CONSTITUTES A "COMMAND STRUCTURE"

A. The Traditional Copyright Principles at Issue in this Case

That copyright law has long afforded protection to non-literal elements of literary and other works flows directly from the Constitution, which secures to authors a limited monopoly in their "writings" – a term interpreted by the Supreme Court to mean works original to the author, possessed of "some creative spark". *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 345-47 (1991) (originality is "the very 'premise of copyright law'") (citation omitted). Consequently, copyright is not limited to literal elements, but can exist in more abstract levels of organization, such as the plot of a novel or play – that is, wherever there is original expression. *See Nichols v. Universal Pictures Corp.*, 45 F.2d 119, 121 (2d Cir. 1930).

The scope of copyright protection includes the original selection, coordination or arrangement of content that may not qualify for protection as individual elements. For example, a "compilation" is statutorily defined as "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". *See* 17 U.S.C. § 101. The Supreme Court has held that even a work "contain[ing] absolutely no protectible written expression, only facts . . . meets the constitutional minimum for copyright protection if it features an original selection or arrangement." *Feist*, 499 U.S. at 348. In other words, the original structure is itself "expression" for purposes of copyright.

Concomitantly, copyright law recognizes limitations on the author's exclusive rights that serve the primary constitutional purpose of "promot[ing] the Progress of Science and useful Arts." U.S. Const. art. I, § 8, cl. 8. One such limitation is the "idea/expression dichotomy", which "'strike[s] a definitional balance between the First Amendment and the Copyright Act by permitting free communication of facts while still protecting an author's expression.'" *Harper & Row, Publishers, Inc. v. Nation Enters.*, 471 U.S. 539 (1985) (citation omitted). By granting a monopoly to authors in their particular way of expressing ideas, but not in the ideas themselves, copyright law encourages the contribution of ideas to the public. Isolating idea from expression to serve the dual constitutional purposes can be a challenging task, particularly for the non-literal aspects of a work, as the trial court's apparent confusion in this case demonstrates. The difficulty arises because, at some level of generality, nothing remains of a work save for the underlying ideas. But between that level of abstraction and the literal expression, there may protectable aspects. *See Nichols*, 45 F.2d at 121.

Another closely related limitation on the scope of copyright protection critical in this case is the exclusion of aspects of works that are functional. This limitation is rooted in the separate regimes for copyright and patent protection enacted pursuant to the Copyright Clause, which authorizes protection for both authors' "Writings", and inventors' "Discoveries". Protection for "functional" aspects of a work is reserved, in appropriate cases, to patent law. Exclusion of both ideas and purely functional aspects of works from copyright protection is explicit in the Copyright Act:

> In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery . . . .

17 U.S.C. § 102(b). This exclusion does *not* mean however that a work that contains some functional aspects is, *ipso facto*, unprotectable by copyright law. Rather, ideas and systems, if novel, that underly functionality are protected by patent, and ways of expressing those ideas and systems, if original, are protected by copyright. *See generally Baker v. Selden*, 101 U.S. 99, 102-03 (1880); *see also Mazer v. Stein*, 347 U.S. 201, 217 (1954) ("Neither the Copyright Statute nor any other says that because a thing is patentable it may not be copyrighted.").

Historically, the U.S. Copyright Office and the U.S. Patent and Trademark Office required authors and inventors to "elect" which form of protection they desired – patent or copyright – if the work in question was eligible for both. In doing so, both agencies were following judicial directives, primarily concerning pictorial, graphic, and sculptural works that also qualified for design patents. Recognizing that copyright and patent protect different attributes of a work, in very different ways, and for different terms, the Copyright Office, in a proceeding initiated at Mr. Oman's direction as Register, issued a new regulation that "[t]he availability of [copyright] protection or grant of protection under the law for a utility or design patent will not affect the registrability of a claim in an original work of authorship." 60 Fed. Reg. 15,605-15,606 (Mar. 24, 1995) (codified at 17 C.F.R. § 202.10).

The next relevant limitation on copyright protection is the merger doctrine. Where there are only one or a few ways to express a given idea, "copying the 'expression' will not be barred, since protecting the 'expression' in such circumstances would confer a monopoly of the 'idea' upon the copyright owner . . . ." *Herbert Rosenthal Jewelry Corp. v. Kalpakian*, 446 F.2d 738, 742 (9th Cir. 1971). This doctrine applies not only to merger of ideas and expression, but equally to merger of processes, methods and other functional aspects of a work with the author's expression. *See, e.g., Gates Rubber Co. v. Bando Chem. Indus.*, *Ltd.*, 9 F.3d 823, 838 (10th Cir. 1993). Importantly for present purposes, there is no merger if the same function can be expressed in multiple ways. *See, e.g.*, *Lexmark Int'l v. Static Control Components*, 387 F.3d 522, 535 (6th Cir. 2004); *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1253 (3d Cir. 1983).

A final relevant constraint is fair use (which is not a limitation on copyrightability, but a defense to infringement), which further balances public and private interests underpinning the Copyright Clause, and provides courts with an equitable tool to determine whether, despite unauthorized copying, mitigating factors suggest that no liability should attach. *See Harper & Row*, 471 U.S. at 560-61. In the context of computer programs, reverse engineering to examine the underlying functionality of software may excuse copying the expression of that functionality. *See, e.g.*, *Sony Computer Entm't, Inc. v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000).

B. <u>The Traditional Copyright Principles Apply to Software</u>

As the trial court acknowledged, the law treats computer programs as "literary works." Decision, at 17. For this reason, the scope of copyright protection for software is governed by the same principles as any other literary work, including all of those outlined *supra*. Courts have accorded protection to literal source and object code, finding it to possess the requisite originality. *See*, *e.g.*, *Whelan Assocs., Inc. v. Jaslow Dental Lab., Inc.*, 797 F.2d 1222, 1234 (3d Cir. 1986); *Lexmark*, 387 F.3d at 533; *see also* Circular 61, United States Copyright Office (instructions for registering source and object code).

Similarly, protection for expressive elements of software reaches beyond literal code to original, non-literal expression.[3] The *Whelan* Court concluded that "copyright protection of computer programs may extend . . . to their structure, sequence, and organization. . ." consistent with legislative intent. 797 F.2d at 1248. That Court reasoned that although the Copyright Act does not use the terms "sequence", "order" or "structure", it is apparent "from the definition of compilations and derivative works, and the protection afforded to them, that Congress was aware of the fact that the sequencing and ordering of materials could be copyrighted . . . ." *Id*. at 1239; *see also Computer Assocs. Int'l, Inc. v. Altai, Inc.*, 982 F.2d 693, 702 (2d Cir. 1992) ("[I]f the non-literal structures of literary works are protected by copyright; and if computer programs are literary works . . . then the non-literal structures of computer programs are protected by copyright"); *Gen. Universal Sys., Inc. v. Lee*, 379 F.3d 131, 142 (5th Cir. 2004) (protection extends "to a program's nonliteral elements, including its structure, sequence,

[3] In 1986, in response to developers' inquiries about registration of software, Mr. Oman, as Register, determined that the Copyright Office needed only one registration per work to cover all copyrightable authorship in the program, including code, screen displays, and structure, sequence, and organization. *See* Circular 61 ("[c]opyright protection extends to *all* the copyrightable expression embodied in the computer program.") (emphasis added).

organization, user interface, screen displays, and menu structures."); *Kepner-Tregoe, Inc. v. Leadership Software, Inc.*, 12 F.3d 527, 535-36 (5th Cir. 1994) (referring to "consensus" that "the structure, sequence, and organization of copyrighted works" may be protected).

Just as protection for computer programs is co-extensive with protection for other literary works, so too do the limitations on protection designed to safeguard the public interest apply to software. Adherence to traditional copyright principles in applying the idea/expression dichotomy to software has been confirmed repeatedly by the courts. *See, e.g.*, *Apple Computer, Inc. v. Microsoft Corp.*, 35 F.3d 1435, 1443 & n.11 (9th Cir. 1994); *Kepner-Tregoe*, 12 F.3d at 533 & n.8; *M. Kramer Mfg. Co. v. Andrews*, 783 F.2d 421, 434 (4th Cir. 1986). The 1979 Commission proposing extension of copyright protection to computer programs emphasized that it was "important that the distinction between programs and processes be made clear." National Comm'n on New Technological Uses of Copyrighted Works, Final Report 18-19 (1979) ("CONTU Report"). That Commission noted that *Baker v. Selden* was "often misconstrued as imposing a limit on the copyrightability of works which express ideas, systems, or processes", but instead "the case properly stands for the proposition that using the system does not infringe the copyright in the description. Congress embodied this rule in §102(b) of the new law." *Id.* at 19.

In the software context, a meticulous analysis may be required to separate protectable expression from unprotectable processes, systems, methods of operation or functionality. As stated in *Johnson Controls, Inc. v. Phoenix Control Systems, Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989), whether "the non-literal components of a program, including the structure, sequence and organization and user interface, are protected depends on whether, on the particular facts of each case, the component in question qualifies as an expression of an idea, or an idea itself." The courts have developed nuanced methods of analyzing non-literal aspects of software in the years since *Whelan*, such as the "abstraction, filtration, comparison" test formulated by the Second Circuit in *Altai*.[4] While the *Altai* Court acknowledged that this test would likely narrow the scope of protection for computer programs, (*see* 982 F.2d at 702), it in no way suggested that non-literal elements of programs would be denied protection simply because they implement or effect functional commands. Indeed, in its definition of "computer program", Congress recognized the "functionality" of software, stating that it "is 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. §101. Contrary to the trial court's conclusion here, nothing in the sophisticated process courts now apply in analyzing

[4] Most Circuits now endorse the *Altai* test. *See, e.g., Atari Games Corp. v. Nintendo of Am. Inc.*, 975 F.2d 832, 839 (Fed. Cir. 1992); *Gates Rubber*, 9 F.3d at 834; *Engineering Dynamics, Inc. v. Structural Software, Inc.*, 26 F.3d 1335, 1342-43 (5th Cir. 1994)).

copyrightability changes the scope of protection accorded non-literal aspects of software, regardless of whether they are complex menu commands, interfaces, or the structure, sequence or organization of a multitude of aspects of those programs.

C. The Trial Court Erred in Determining that the Original, Expressive Structure of Oracle's Program Was Not Protectable Solely Because It Constitutes a Command Structure or "Method of Operation"

Contrary to these traditional principles, the court below held that the structure, sequence and organization of the Oracle program was not protectable by copyright because it constituted a "command structure." In essence, the court relegated the organization of the Java software packages exclusively to patent law, eschewing even consideration of whether or not the non-literal aspects were also entitled to copyright protection.

From Mr. Oman's perspective, the rejection of copyright protection for the organization of Oracle's software packages threatens to do violence to the very concept of copyright protection of software. In examining whether Google was free – in the court's words – to "organize its Android methods under the same class and package scheme as in Java" (Decision at 36), the court made key findings that illustrate why its conclusions flout basic principles of copyright law. First, it acknowledged that the structure, sequence and organization of the Java methods are "creative" and "original" and thus expressive. It also found that "the rules of Java did not insist that these methods be grouped together in any particular class….

Oracle is entirely correct that the rules of the Java language did not require [] the same grouping pattern (or even that they be grouped at all, for each method could have been placed in a stand-alone class)." *Id.* at 36. As a result, there "was nothing in the rules of the Java language that required that Google replicate the name groupings even if Google was free to replicate the same functionality." *Id.* at 37. Thus, according to the court's own reasoning, Oracle's complex and creative structure, sequence and organization are sufficiently original to qualify for protection and do not violate the merger doctrine. This conclusion should have ended the court's copyrightability analysis.

Instead, the court declined to extend copyright protection to the "overall scheme of file name organization", because it is "a command structure for a system or method of operation of the application programming interface", made up of "a long hierarchy of over six thousand commands to carry out pre-assigned functions." Decision at 37. The trial court seemingly adopted the First Circuit's approach in *Lotus Development Corp. v. Borland International, Inc.*, 49 F.3d 807, 815 (1st Cir. 1995), *aff'd by an equally divided Court*, 516 U.S. 233 (1996), that a menu command hierarchy is uncopyrightable under § 102(b) as a "method of operation", defined as "the means by which a person operates something." This approach represents far too cramped a definition of a "method of operation" –

indeed, this definition is nearly synonymous with the Act's definition of "computer program", cited above.

Taken to the highest level of abstraction, all computer programming, including source code, can be characterized as a "method of operation" or system of commands, yet no one would argue that none of it is protected by copyright, as that would be flatly contrary to the Act and the intent of Congress. A number of circuit courts have recognized the fallacy of the approach embraced by the court below. "That the words of a program are used ultimately in the implementation of a process should in no way affect their copyrightability." *Apple v. Franklin Computer*, 714 F.2d at 1251. *See also Mitel, Inc. v. Iqtel, Inc.*, 124 F.3d 1366, 1372 (10th Cir. 1997) (although an element may be "characterized as a method of operation, that element may nevertheless contain expression that is eligible for copyright protection. Section 102(b) does not extinguish the protection accorded a particular expression of an idea merely because that expression is embodied in a method of operation at a higher level of abstraction."); *Apple Computer, Inc. v. Formula Int'l, Inc.*, 725 F.2d 521, 523-24 (9th Cir. 1984) (rejecting notion that operating systems are "processes" and thus unprotectable).

Nor does the fact that a structure is made up of individually unprotectable commands distinguish it from a compilation of individually unprotectable facts, selected and organized in a sufficiently original way to qualify for protection.[5] 17 U.S.C. § 101. Similarly (and consistent with the trial court's analogy to libraries, Decision at 5), the Dewey Decimal Classification system, "the most widely used classification system in the world" (www.oclc.org/dewey), is "made up of ten classes, each divided into ten divisions, each having ten sections", and is hierarchical. The system itself is functional – it assists in organizing and locating books – and the underlying classification number for each book is not protectable, yet the system as a whole was protected historically by copyright. Likewise, choreography – the arrangement and sequences of movements, steps and patterns of dancers – is subject to copyright protection (17 U.S.C. § 102(4)), even though the underlying movements and steps are not protectable; the options for each movement or step are limited, and each command serves the function of directing the movement of the dancer.[6] These examples illustrate that simply determining

[5] The trial court rejected protection for Oracle's declaring code, relying on a Copyright Office regulation indicating that "words and short phrases" may not be registered. *See* Decision at 14-15, 35-36 (*citing* 37 C.F.R. § 202.1(a)). This regulation is wholly inapposite to whether a compilation, or the structure, sequence and organization of thousands of lines of code, is protectable. In Mr. Oman's experience, the Copyright Office examines any particular "short phrase" (including individual lines of code) on a case-by-case basis to determine whether it possesses a sufficient "creative spark" to warrant protection. For example, consider the Ogden Nash poem "Candy is dandy, but liquor is quicker"; although only seven words, it almost surely qualifies for protection.

[6] Other examples include blueprints, instruction manuals, dictionaries, encyclopedias, maps, and the SAT test, *see Am. Dental Ass'n v. Delta Tental Plans Ass'n*, 126 F.3d 977, 978 (7th Cir. 1997), and CONTU noted that "games and

that non-literal elements of a work serve a functional purpose, or constitute a system or series of commands, is insufficient to determine the elements' protectability.

Consequently, although the individual components of Oracle's program may (or may not) be methods of operation, the original and creative structure, sequence, and organization of them can nonetheless be the subject of copyright protection. The court below did ***not*** hold that the structural components of Oracle's packages merged with expressive content and were therefore unprotectable. Instead, the court held that that structure was not protectable because, although it "resembles a taxonomy, it is *also* a command structure." Decision at 37. In other words, although the structure is expressive, it is also functional. The court apparently believed that it had to choose between the two, but in light of its finding that there were multiple ways to express the same command structure, that was simply incorrect. The cases teach that, at a lower level of abstraction, an expressive structure, or taxonomy, or method of operation may be protected separate and apart from its functional components.

By failing to recognize this, the court essentially contravened the congressional and judicial mandates which further the constitutional bargain of

game-playing" are "closely analogous" to software, because "one may not adopt and republish or redistribute copyrighted game rules, but the copyright owner has no power to prevent others from playing the game". CONTU Report 20.

extending protection to all creative, original aspects of works of authorship in exchange for contribution of the underlying ideas and functionality (and when protection lapses, the expression itself) to the public. Rather than applying the well-established limitations that balance the dual constitutional goals, the court enacted its own *sui generis* functionality exception to the scope of copyright protection for software, and attempted to buttress that ruling by invoking an inverted concept of software interoperability, as discussed below.

III.

THE COURT ELEVATED "INTEROPERABILITY" CONCERNS TO AN UNWARRANTED PLACE OF PRIMACY IN THE COPYRIGHT LAW

A. That Aspects Of Software Implicate "Interoperability" Does Not Predetermine That It Is Unprotectable

At the heart of the trial court's discussion of copyrightability is its conclusion that the Java command structure is "interoperable" with third-party software, and therefore non-protectable. Under traditional copyright principles, software that incorporates interoperability should be examined just as any other software – indeed, just as any literary work – in order to determine whether it contains creative expression separable from the underlying enabling function. The court in this case seemingly concluded that if software enables interoperability, even if expressed in an original way, by definition it merges with its underlying functionality. This conclusion appears based on a fundamental misunderstanding of prior case law discussing one aspect of interoperability. The specific interoperability concerns raised by the court are relevant, if at all, only in connection with a fair use defense, not to determine the threshold question of copyrightability.

In *Altai*, the Second Circuit identified a number of possible "filters" that might be used to identify non-copyrightable aspects of a computer program, including structures dictated by efficiency, those found in the public domain, and "external factors" circumscribing a programmer's design choices. 982 F.2d at 709-10. In light of the constitutional stricture that originality is the *sine qua non* of copyright, *see Feist*, 499 U.S. at 345, relevant external factors are those that ***require*** a programmer to write the program in a specified way, rather than exercising judgment and creativity in choosing from a range of options. One external factor identified in *Altai* was "compatibility requirements of other programs with which a program is designed to operate." 982 F.2d at 710. The Second Circuit did not even suggest, much less hold, that all software components enabling interoperability would so limit a programmer's choices that they merged with expression and are automatically uncopyrightable. Nor did it suggest that because interoperable components are "functional",[7] they can never contain protectable expression.

To understand what *Altai* meant by "compatibility," it is crucial to focus on which software is being examined. Because the question is the copyrightability of *plaintiff's* program, the Second Circuit correctly focused on that program and the extent to which *plaintiff*-programmer's choices were strictly dictated by the need to ensure that it could interact with existing third-party programs. The focus was not, nor should it have been, on the third-party software or hardware creator's need to interact with plaintiff's program or any other existing program. Logically, and as a matter of law, the functionality of third-party products has nothing to do with whether plaintiff created software in the first instance that contains sufficient originality of expression to be protected by copyright law. The needs of the third party programmer are relevant, if at all, only in the infringement context – specifically in the fair use analysis – but they have absolutely nothing to do with protectability *ab initio*.

This issue was addressed in an early computer software copyright case from the Third Circuit which the court below failed to mention, *Apple v. Franklin*

[7] Software concerned with interoperability is no more or less "functional" than code that enables a program to display images on a monitor or send email; it is a string of commands resulting in a series of actions. *See Apple v. Franklin Computer*, 714 F.2d at 1251 (rejecting argument that, while application programs were protectable, operating systems were not, because "[b]oth types of programs instruct the computer to do something").

Computer. There, defendant argued that compatibility considerations of its software rendered plaintiff's operating system uncopyrightable and that there were "a limited number of ways to arrange operating systems to enable a computer to run the vast body of Apple-compatible software." The Court squarely held that this argument had "no pertinence to either the idea/expression dichotomy or merger," because "[t]he idea which may merge with the expression, thus making the copyright unavailable, is the idea which is the subject of the expression." 714 F.2d at 1251. While defendant "may wish to achieve total compatibility with independently developed application programs written for the Apple II, [] that is a commercial and competitive objective which does not enter into the somewhat metaphysical issue of whether particular ideas and expressions have merged." *Id.* at 1253. This remains a correct statement of the law, cited by the Second Circuit in *Altai*, before addressing circumstances where external considerations *might* matter, *i.e.*, when they might affect choices made by *plaintiff's* programmer.

This exclusive focus on the programming choices of the party claiming copyright protection was embraced by this Court in *Atari Games Corp. v. Nintendo of America Inc.*, 975 F.2d 832 (Fed. Cir. 1992), a case involving software that prevented other game cartridges from being used in a console. The Federal Circuit held that the organization and sequencing of that program constituted expression – "creative organization and sequencing unnecessary to the lock and key function" –

which did not merge with the function, since there were multiple ways to create the expression.[8] *Id.* at 840. This Circuit treats these aspects of software – that enable interoperability but are nonetheless expressive – exactly the same way as it treats any other work for copyrightability: it examines them to determine whether there is creative expression that does not merge with the function.

In considering the copyrightability of command codes in *Mitel v. Iqtel*, the district court first properly considered external factors influencing creation of the work, but then erred, according to the Tenth Circuit, by shifting to a discussion of "whether external factors such as market forces and efficiency considerations justified Iqtel's copying of the command codes," rather than "remain[ing] [focused] upon the external factors that dictated Mitel's selection of registers, descriptions, and values." 124 F.3d at 1375 (original emphasis omitted).

The court here made a similar mistake by focusing on the interoperability of Google's software, not Oracle's software. Although it states that "interoperability is at the heart of [Oracle's] command structure" (Decision at 38), it never analyzes the Oracle software at all for this purpose. The court makes only passing reference to the Oracle packages needing to be grouped a particular way in order to facilitate interoperability (Decision at 12), but that is not interoperability or compatibility in

[8] The *Atari* Court explained that although "lock out" codes were clearly functional, Nintendo chose "arbitrary programming instructions and arranged them in a unique sequence to create a purely arbitrary data stream," which was protectable. 975 F.2d at 840.

the *Altai* sense of driving programmer choices – that is mere standardization of Oracle's own software organization. If anything, choosing a particular way of expressing the underlying functionality from among many *supports* a conclusion that Oracle is seeking protection only for its own particularized expression. The fact that this standardization facilitates later development of compatible software by others is irrelevant to the question of copyright in the original program.

The court then compounded its error by concentrating not on Android's interoperability with Oracle's software, but Android's interoperability with software made by third parties. The court noted that other programmers, not working for Oracle (or Google), had written programs for the Java platform "and necessarily used the command structure of names at issue." Decision at 38. What the court meant by "used" in this context is unclear, but as long as those programmers did not copy Oracle's protected expression in doing so (or, if they did, either had a license or made fair use of some of the Java expression), the programs would not be infringing. The court held that simply because ***Google*** wanted to create a mobile platform *on which the third-party software would operate*, the original Oracle command structure was interoperable and could not be protected. *Id.* That is wrong both as a matter of law, for all the reasons set forth above, and also as a matter of policy.[9]

The public policy implications of this wholesale inversion of the proper analysis to determine protectability of software are disturbing. In the first instance, this inversion does violence to the constitutional mandate to "secur[e] . . . to Authors . . . the exclusive Right to their [] Writings . . . ." Second, it could have *ex post facto* effect: under the court's reasoning, even if the Oracle command structure was copyrightable at the time it was created, the later development of a market based around the Java platform could somehow vitiate protection before the end of the copyright term. The fact is that even if the specific market considerations that the court posited somehow existed at the time the software was first authored, it would still have been inappropriate to consider them, as they could not have affected the programmers' creative choices.

B. The Court Improperly Introduced Commercial Considerations Only Appropriate to a Fair Use Analysis Into the Examination of Copyrightability

In lieu of following the mandate of *Apple v. Franklin Computer* and *Atari* that it is the interoperability needs of plaintiff, not defendant, that control the

[9] It also appears to be wrong as a matter of fact: Oracle points to admissions by Google that Android was not designed to be "interoperable" with Java applications at all. Br. at 65-66. Assuming this is correct, the court's decision has a decidedly anti-competitive effect: While denying copyright protection to the creator of an original work, it secures a plagiarist's use of non-interoperable software.

copyrightability analysis, the court below cited two Ninth Circuit cases to find support for its *sui generis* approach to interoperability, *Sega Enterprises Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992) and *Sony Computer Entertainment, Inc. v. Connectix Corp.*, 203 F.3d 596 (9th Cir. 2000). While the court acknowledges that these were fair use cases in which copyrightability was only addressed tangentially (Decision at 29), the court failed to appreciate how this important distinction rendered its reliance on those cases totally inapposite.

In *Sega*, Accolade made intermediate copies of Sega's software in order to reverse engineer it, to understand how Sega's game console interfaced with Sega's own game cartridges. The court below cited *Sega* for the proposition that "interface procedures for compatibility were functional aspects not copyrightable under Section 102(b)" which is unquestionably true, as "procedures" cannot be copyrighted. But in *Sega*, the ***only*** question was whether Accolade's intermediate copying was a fair use; the Ninth Circuit never reached the issue of whether or not the software code that incorporated the functional aspects that Accolade sought to mimic also contained separable creative expression. It never reached the issue of whether or not there was protectable expression in the interface procedures, notwithstanding the functional aspects of the program.

The court below places significant reliance on a footnote in *Sega* saying that one component of the Sega interface procedures, a "lock out" code analogous to the one in *Atari*, was not protectable. But *Sega* was careful to distinguish the difference between these lock out codes: while "[c]reativity and originality" were found present in the Atari code, the Sega code was short and unvarying and was the sole means of unlocking the console. 977 F.2d at 1524 n.7. In other words, the *Sega* Court distinguished between code that could have been expressed in many ways (indicating that the particular version Atari programmers chose was expressive), and code for which there had been "no showing of a multitude of different ways" to express the underlying function. *Id.* In this case, the trial court conflated "unnecessary functionality" with separable expression based on this footnote, "as drawing a line between copying functional aspects necessary for compatibility (not copyrightable) versus copying functional aspects unnecessary for compatibility (possibly copyrightable)." Decision at 31. This purported distinction between copying two categories of "functionality" represents a significant departure from established case law. Under copyright, only copying of expression is germane; duplication of functionality alone is always permissible, absent protection.

The court's reliance on *Sony* is equally misplaced. There, the Ninth Circuit only determined, once again in the fair use context, that Sony's program – like all software – contained functional aspects; it never attempted to identify whether it also had expressive aspects. Based primarily on these two fair use cases, the court below concluded that the Ninth Circuit "plac[es] a premium on functionality as indicating uncopyrightability." Decision at 32 n.6. That conclusion is simply in error. In both *Sega* and *Sony*, since defendants made intermediate copies solely to understand the underlying functionality in order to create a new and different product, the purpose of the copying was fair. *See Sega*, 977 F.2d at 1522-23; *Sony*, 203 F.3d at 606-07. Here, the undisputed facts show that Google first tried to license the Java platform or partner with Oracle and, unable to reach agreement, created its own software. Decision at 5-6.

In essence, the decision below judicially creates a free compulsory license allowing anyone to avoid the exclusive rights of copyright simply because they need to use the expressive aspects of the copyrighted work to create an interoperative derivative work, even if the second work does nothing to transform the first, and even if it adversely affects the market for licensed derivatives. While on first blush, this appears to be pro-competitive, it is not. By protecting a single way of writing a program or modeling its structure, sequence and organization, copyright law encourages other creators to develop new ways of writing it or modeling the same underlying function, which may in fact be better. That promotes the public interests identified in the Copyright Clause far more effectively than chilling development by reducing the range of copyright protection accorded software.

The trial court's decision appears driven, ironically, by the popularity of Oracle's Java platform and by the court's judgment that the market needs free rein (without the inconvenience of proving that a particular use is fair) to create interoperable or compatible software. Several passages in the court's Order betray a bias towards free development of software and interoperability as a paramount and overriding consideration.

Also underlying the decision is an apparent belief that patent law should be the primary mechanism of protection for software. *See, e.g.*, Decision at 32 (noting recent increase in the number of software patents in the United States, and suggesting the CONTU Report is out-of-date); at 33 (citing suggestion by commentator that structure, sequence and organization of computer programs "will become a mere incident to the patentable idea"); at 33 (noting that Oracle has patents covering some aspects of Java); and at 35 (stating that by seeking protection here, Oracle sought to "bypass this entire patent scheme and claim ownership over any and all ways to carry out methods for 95 years"). But that is not the case, as discussed above: different aspects of software are protected under the different regimes. As the CONTU Report noted, "The purpose of copyright is to grant authors a limited property right in the form of expression of their ideas. The other methods used to protect property interests in computer programs have different conceptual bases and, not surprisingly, work in different ways. . . .

Patents are designed to give inventors a short-term, powerful monopoly in devices, processes, compositions of matter, and designs which embody their ideas." CONTU Report 16 (1979).

The court's reluctance to grant a "95 year monopoly" seems to stem not just from believing it had to choose between copyright and patent protection, but also from mistakenly equating the copyright and patent monopolies. The copyright monopoly is longer, but much more limited than the short, "powerful" patent monopoly. An inventor can bar a subsequent inventor from practicing an invention, even if the second inventor had no knowledge of the first and independently created the same invention, and there is no "fair use" defense to patent infringement. Not so on the copyright side. Had Google reverse engineered the programming packages to figure out the ideas and functionality of the original, and then created its own structure and its own literal code, Oracle would have no remedy under copyright whatsoever. For financial reasons, Google chose not to follow this route.

If other courts were to adopt the rationale of the district court in this case, many of the important expressive features of software would effectively lose meaningful copyright protection. The United States spent twenty years trying to convince its trade partners and the hierarchy of the World Intellectual Property Organization to fully protect software under the umbrella of copyright in

GATT/TRIPs and the WIPO Copyright Treaty. It would be ironic if that historic achievement was totally subverted by the faulty reasoning of the court below. Even more to the point, if such a radical departure from current law and treaty obligations were to be made, that would be exclusively within the purview of Congress.

Date: February 19, 2013

By: /s/ Marcia B. Paul
Marcia B. Paul
Davis Wright Tremaine LLP
1633 Broadway
New York, NY 10019
212-489-8230
Attorney for Amicus Curiae Ralph Oman

CERTIFICATE OF COMPLIANCE
UNDER FEDERAL RULES OF APPELLATE PROCEDURE
32(a)(7) AND FEDERAL CIRCUIT RULE 32

Counsel for Amicus Curiae Ralph Oman certifies that the brief contained herein has a proportionally spaced 14-point typeface, and contains 6959 words, based on the "Word Count" feature of Word 2010, including footnotes and endnotes. Pursuant to Federal Rule of Appellate Procedure 32(a)(7)(B)(iii) and Federal Circuit Rule 32(b), this word count does not include the words contained in the Certificate of Interest, Table of Contents, and Table of Authorities.

Date: February 19, 2013 By: /s/ Marcia B. Paul
Attorney for Amicus Curiae Ralph Oman

CERTIFICATE OF SERVICE & CM/ECF FILING

13-1021, -1022

I hereby certify that I caused the foregoing Brief for Amicus Curiae Ralph Oman Supporting the Position of Plaintiff-Appellant and Urging Reversal to be served on counsel via Electronic Mail generated by the Court's electronic filing system (CM/ECF) with a Notice of Docket Activity pursuant to Appellate Rule 25 and the Federal Circuit's Administrative Order Regarding Electronic Case Filing ECF-6.

I certify that an electronic copy was uploaded to the Court's electronic filing system.

/s/ Ramiro Honeywell
Ramiro Honeywell