A Brief Guide to Rights in Computer Software: Recent US Decisions

by William B. Bierce

The development and marketing of computer software have always faced commercial and financial risks. Recent judicial decisions interpreting the law on intellectual property rights in user interfaces and other 'non-literal elements' of computer software impose additional uncertainties. This article is intended to update the business manager, the software developer, the multinational corporation (as both user and developer) and investors and financiers in intellectual property rights issues.

Investors and business managers should consider their response to these developments. Strategic responses include pre-litigation strategies, lobbying for possible legislation to revise the rules governing rights in software, and anticipation of a possible definitive US Supreme Court interpretation of the interplay of different intellectual property rights. After reviewing these decisions, certain international and foreign legal considerations are discussed briefly.

Patent Law

Patent law provides 17 years exclusive right to use an idea which is utilitarian, novel, not obvious to one skilled in the art, and comprises in integral element in a useful process. Useful processes incorporating a mathematical algorithm may be patented if the mathematical algorithm is applied in one or more steps of an otherwise statutory 'process' claim, or one or more elements of a statutory 'apparatus' claim.[1] Recently, software developers have attempted to establish a large number of patents in small improvements to software and hardware technology. Such a pool of patents is then used to negotiate royalties or technology exchange agreements with competitors.

For the small developer, the threat of a patent litigation can be devastating. The first one to discover a process is rewarded with exclusive rights. Subsequent discoverers, even if they had no access to the invention of the first discoverer, are precluded from making, using or selling any apparatus embodying the patented invention, or using the patented process.

Surprisingly, copyright law has become so complex and unpredictable a solution that one appellate court suggested recently that patent law should be the standard, not copyright law.[2]

Copyright

Recent judicial decisions narrow the scope of copyright protection for utilitarian literary works over a century ago. In *Baker v Selden*[3] an author had copyrighted a book containing lined pages with headings intended to illustrate the manner in which the author's bookkeeping system operated. The author's representative sued another author, who had used the lined pages, for copyright infringement. The Court denied copyright protection, concluding that the lined pages with headings were blank forms which defined an idea, not merely an expression. In addition, the Court concluded that those elements of the work which 'must necessarily be used as incident to 'the idea, system or process that the work describes, are also not protected by copyright.[4]

Currently, at least four tests exist to determine whether non-literal elements of a computer program are protectable by copyright.[5] The first one looks at the 'structure, sequence and organisation:' of a program to distinguish 'idea' from 'expression'. The second applies a more analytical approach, call 'analytic dissection'. A third compares non-literal elements at different levels, called 'abstraction-filtration-comparison. The fourth examines 'functionality'.

These different tests co-exist, each one being applicable as binding precedent in a different judicial district of the United States. The last three are similar and compatible, but apply different methodology.

Structure, Sequence and Organisation

Under the 1986 decision of *Whelan v Jaslow*[6] once it has been determined what is 'the idea' of a given program as a whole, then the 'structure', sequence and organisation' are copyrightable forms of expression. The *Whelan* court defined this test:

> 'The line between idea and expression may be drawn with reference to the end sought to be achieved by the work in question. In other words, the purpose or function of a utilitarian work would be ***the work's idea and everything that is not neces-***

> *sary to that purpose or function would be part of the expression of the idea.* ...Where there are various means of achieving the desired purpose, then the particular means chosen is not necessary to the purpose; hence, there is expression, not idea.'[7]

In short, 'the purpose of function of a work or literary device is part of that device's 'idea' (unprotectable portion).'[8]

While followed by several courts,[9] this test has been rejected by other courts[10] and commentators. *Whelan* focused only on one single idea as constituting 'structure, sequence and organisation'. In fact, nonliteral elements in programs consist of both 'static structure' and 'dynamic structure.' Moreover, the *Whelan* solution failed to consider the interplay of each subroutine, and each subsubroutine, which itself could be either an idea or an expression.

Analytic Dissection

The judicial concept of 'analytic dissection' was originally developed as the 'extrinsic' part of a two-part test of substantial similarity between two works of authorship. The 'intrinsic' test examines the response of the ordinary reasonable person. The 'extrinsic' test compares the expressions of ideas embodied to two works. Only recently has the 'extrinsic' analytic dissection portion of this approach been used to determine the scope of copyright protection.[11]

In *Brown Bag Software v Symantec Corp.*[12] the appeals court approved of the use of 'analytic dissection' as relevant not only to the copying element of a copyright infringement claim, but also to the ownership of such a claim. The lower court had compared specific screens, opening menus and keystroke. Analytic dissection involves three steps:

(a) identification of discrete non-literal elements, such as specific screens, keystrokes, etc;

(b) determination of whether similarities between the two programs result from unprotectable or unprotected expression; and

(c) where the two works are found to be similar without regard to the scope of the copyright in the plaintiff's work, the source of similarity must be identified, and a determination must be made whether this source is covered by the plaintiff's copyright.

"...the court applied a new definition of the scope of copyright protection for nonliteral elements of the software"

Abstraction-Filtration Comparison

A more complex analysis, based on a hieratical sequence of analysis, was used in *Computer Associates International Inc. v Altai, Inc.* In *Computer Associates*, the US Court of Appeals for the Second Circuit decided that Computer Associates could collect damages for copyright infringement from the employer of a former CA employee who had, without the employer's knowledge, misappropriated portions of CA source code in developing a 'common system interface'. The program was designed to permit the employer to port its job scheduling software across various hardware platforms by using a standard set of procedure calls.

Once the employer discovered the misappropriation, it invested six man-months of work in redesigning the software without any contact with the tainted former CA employee. This redevelopment effort resulted in a non-infringing software product, according to the court.

In reaching this result, the court applied a new definition of the scope of copyright protection for nonliteral elements of the software. The literal elements - that is, source code and object code - are already protected by copyright.[13] Certain types of screen displays, which constitute audiovisual works protected under a statutory category other than 'literary works', are excluded from this analysis.[14]

Since copyright protection extends only to the expression of an idea, and not to the idea itself,[15] the judge's initial inquiry starts with the distinction between the idea and its expression. Such an inquiry is inevitably *ad hoc* with no fixed boundary.[16] The inquiry is complicated by the dual character of software, being both creative (artistic) and technical (utilitarian).

The *Computer Associates* court adopted a three-step procedure to determine whether the non-literal elements of two or more computer programs are 'substantially similar'.

> 'In ascertaining substantial similarity under this approach, a court would first break down the allegedly infringed program into its constituent structural parts. Then, by examining each of these parts for such things as incorporated ideas, expression that is necessarily incidental to those ideas, and elements that are taken from the public domain, a court

> would then be able to sift out all non-protectable material. Left with a kernel, the court's last step would be to compare this material with the structure of the allegedly infringing program. The result of this comparison will determine whether the protectable elements of the programs at issue are substantially similar so as to warrant a finding of infringement'[17]

This 'abstractions' test resembles reverse engineering on a theoretical level. The court is directed to 'dissect the allegedly copied program's structure and isolate each level of abstraction contained within it'.[18] 'A program has structure at every level of abstraction at which it is viewed.'[19] The appellate court approved the lower court's use of a progression (in levels of abstraction) in increasing generality, from object code, to source code, to parameter lists, to services required, to general outline.[20]

Once the abstraction of levels is established, then the process of 'successive filtration' would apply the 'idea-expression' test to determine whether, at each level of abstraction, there is idea or expression. Three non-protected non-literal elements must be identified:

(a) those which are dictated by considerations of 'efficiency'; namely, the degree to which the choices in writing the set of modules in the program are so narrowly confined that only one or two forms of expression are workable options.[21] (If the use of this particular set of modules is dictated by the necessity efficiently to implement that part of the program's process, then the expression is 'merged' into the idea and is not protected by copyright[22]);

(b) those which necessarily incidental to the idea and

(c) those which are required by factors external to the program itself, such as the mechanical specifications of the computer, compatibility requirements of other programs intended to operate concurrently, computer manufacturer's design standards (such as the Apple interface), demands of the industry for which a vertical application is developed, and widely accepted programming practices[23] within the computer industry.[24]

"*The Computer Associates decision represents an explicit attempt to limit the claims of the copyright holder ...*"

The *Computer Associates* court rejected the criticisms of various non-parties which had submitted legal conclusions to assist the court. The principal criticism was that such a process would prove a disincentive to the investment of time, energy and funds required to design and improve program structures. The basic purpose of the Copyright Act was noted as promoting the public welfare, not private benefit, but that short-term private benefit was an element in achieving such public welfare over time, as noted in a Supreme Court decision[25], rendered after the *Whelan*, decision, rejecting the assumption implicit in *Whelan* that copyright law is intended to reward substantial effort.

The *Computer Associates* decision represents an explicit attempt to limit the claims of the copyright holder to protection for elements which might be useful and novel in the computer industry. In that sense, the court almost admitted it would have liked to apply patent law concepts of novelty and non-obviousness.

Indeed, it may well be that the Copyright Act serves as a relatively weak barrier against public access to the theoretical interstices behind a program's source and object codes. This results from the hybrid nature of a computer program, which, which it is literary expression, is also a highly functional, utilitarian component of the larger process of computing.

Generally, *we think that copyright registration, with its indiscriminate availability, is not ideally suited to deal with the highly dynamic technology of computer science.* Thus far, many of the decisions in this area reflect the courts' attempt to fit proverbial square peg into a round hole.[26] [Emphasis added].

The court then urged the use of patent law as a supplement or alternative, since patent law requires novelty from the beginning and non-obviousness of the process.

'Essential Elements' Test

In 1990, in *Lotus Dev. Corp v Paperback Software International.*[27] US District Judge Robert Keeton in Boston established a method for unravelling, or depechage, of elements which are essential to expressing an idea (or other non-copyrightable system. process, procedure or method). In a sequel[28] Judge Keeton modified this standard to take into account the statutory list of other non-copyrightable elements. This test was used in reviewing Borland's use of Lotus's menu system:[29]

First, in making the determination of 'copyrightability', the decision maker must focus upon alternatives that counsel may suggest, or the court may conceive, along the scale from the most generalised to the most particularised and choose some formulation, some conception of the 'idea', 'system', 'process', 'procedure' or 'method' for the purpose of distinguishing between the idea, system, process, procedure, or method and its expression.

Second, the decision makers must focus upon whether an alleged expression of the idea, system, process, procedure or method is limited to the elements essential to expression of that idea, system, process, procedure or method (or is one of only a few ways of expressing the idea, system, process, procedure, or method) or instead includes identifiable elements of expression not essential to every expression of that idea, system, process, procedure or method.

Third, having identified elements of expression not essential to every expression of the idea, system, process, procedure or method, the decision maker must focus on whether those expressive elements taken together, are a substantial part of the allegedly copyrightable 'work'.[30]

This standard differs substantially from the *Whelan* 'structure, sequence and organisation' test, but is compatible with the *Brown Bag* 'analytic dissection' test and the *Computer Associates*' abstraction-filtration-comparison' test.

In the *Lotus* case, Borland's Quattro Pro was found to infringe Lotus 1-2-3 in various ways relating to the screen displays, menus and macros.

Borland admitted that its:

> 'employees reviewed books about [Lotus] 1-2-3, Release 2.01, written by third parties, which books contain schematic or menu-free type representations of the 1-2-3 menu command hierarchy. Borland used these third-party menu frees to construct 1-2-2-compatible menu hierarchies in their own products.'[31]

"These recent American decisions and the EC Software Directive reduce the scope of copyright protection in the non-literal elements of software"

The court ruled that Borland's admission that the 'Quattro programs duplicate the set of 'functional relationships' of Lotus 1-2-3 and were designed to do so is conclusive against Borland on the issue of copying that set of functional relationships[32] Borland admitted that it had intentionally incorporated into its user interface the 1-2-3 menu commands and menu command hierarchy, as well as the Lotus 1-2-3 keystroke sequences and macro language. Borland's indirect copying, by reference to third-party sources describing Lotus 1-2-3, did not protect Borland from a claim of copyright infringement. The only non-literal element which Borland did not clearly copy was the long prompts, as to which a jury trial may be held.

Infringement arose, said the court, since 'reasonable jury could find that the menu command hierarchy was limited to one or even several alternate designs at the time it was created.'[33] As a result, there were several forms of expression for this particular idea, and Lotus' form was a protectable expression.

Interoperability and Other International Implications

The European Community ('EC') Software Directive[34] permits any person to reverse engineer any software in order to:

> 'observe, study or test the functioning of the program in order to determine the ideas and principles which underlie any element of the program' in conjunction with a permitted use of the program.'[35]

and to

> 'obtain the information necessary to achieve the interoperability of an independently created computer program with other programs, provided... (b) the information necessary to achieve interoperability has not previously been readily available [to that person]; and (c) these acts are confined to the parts of the original program which are necessary to achieve interoperability.'[36]

American courts have not had to confront the impact of such mandates on the scope of copyright protection. Indeed, recently a federal district court in California observed rejected the 'public policy' argument, that 'immediate copying' should be allowed in order to permit interoperability.[37]

These recent American decisions and the EC Software Directive reduce the scope of copyright protection in the non-literal elements of software. Both express an intention to reduce copyright litigation involving programming techniques dictated by efficiency, factors external to the program itself of taken from the public domain.

Whether the US Congress or the courts will extend this restrictive interpretation to achieve easier interoperability remains to be seen. Certainly, informed market participants should follow carefully such developments as well as the impact of future European judicial decisions on copyright and marketing strategy in the United States.

Trade Secrets

Definition and Scope.

The common law protects trade secrets. The scope and conditions such protection are limited by various requirements.

A trade secret may consist of any formula, pattern, device or compilation of information which is used in one's business, and which gives him an opportunity to obtain an advantage over competitors who do not know it.[38]

In determining whether a trade secret exists, various elements must be considered. Under New York law, these elements include:

(a) the extent to which the information is known outside his business;

(b) the extent to which it is known by employees and others involved in his business;

(c) the extent of measures taken by him to guard the secrecy of the information;

(d) the value of the information to him and to his competitors;

(e) the amount of effort or money expended by him in developing the information; and

(f) the ease or difficulty with which the information could be properly acquired or duplicated by others.[39]

Trade secret protection extends not only to source and object codes, but also to the manner in which several non-secret utility programs are arranged to create a computer software product.[40] 'A trade secret can exist in a combination of characteristics and components, each of which, by itself, is in the public domain, but the unified operation of which, in unique combination, affords a competitive advantage and is a protectible secret', where that particular combination is neither obvious nor easily duplicated.[41]

> ***"Software developers and users should also be sensitive to the need to protect other trade secrets"***

Ironically, the *Computer Associates* 'abstraction-filtration-comparison' method of determining copyrightability filters out elements which are in the public domain, such as programming techniques which, if not standard, then are at least 'commonplace' in the computer software industry.[42] If a unique combination of elements in the public domain is an integral part of a program, then they are not entitled to copyright (under such analysis) but may be entitled to trade secret protection, provided that the particular combination is not obvious and is not easily duplicated. The 'abstraction-filtration-comparison' approach therefore ignores the protection afforded to compilations of information in the public domain, thus contradicting an essential element of copyright law. The conflict of these two rules, one from copyright law, the other from trade secret law at common law, may effectively prevent software developers from enjoying any protection for user interfaces where other forms of intellectual property protection are not available.

Software developers and users should also be sensitive to the need to protect other trade secrets. Protected trade secrets may include features such as the equations and formulae used in a cost-estimating program, its architecture (the identification and choices made between various alternatives in a decisions free) and the confidential feedback which customers provide with regard to enhancements which they desire.[43]

Techniques

Developers should adopt a systematic method for identifying and protecting the secrecy of their confidential business information. The methods used by Computer Associates International were recently found sufficient by a federal court. These methods include:

(a) employment agreements: each new employee signs an employment agreement in which he or she agrees not to use or misappropriate any trade secret or confidential information for his own benefit or for the benefit of any other person or entity;

(b) 'exit' agreements: departing employees are asked to sign 'departing employee agreements', in which they reiterate these undertakings and agreed to destroy or return all confidential information;

(c) nondisclosure agreements: all persons to whom a software product was shown, such as prospective distributors, were asked to sign a nondisclosure agreement in which they acknowledge receiving confidential and proprietary information and agree not to reverse engineer or otherwise attempt to use it;

(d) license agreements: all licensees agreed not to remove or destroy any proprietary markings, not to make copies (except for archival purposes), to maintain the confidentiality of the programs and not to disclose them, not to decompile or reverse engineer, and to return or destroy and delete all copies upon termination of the license. Licenses were for object code only;

(e) proprietary manuals: all manuals were marked as 'proprietary and confidential', expressly prohibited copying without written permission, and required all copies to include the same legend; and

(f) office security measures: identification badges, restricted access to the building, supervision of all visitors in the building, night guards and triplicate storage of source codes (at headquarters, at an escrow site and with the developers who work on the product).

Fiduciary Obligations

Written employment agreements normally establish relationships of trust and confidence, imposing a duty on the employee not to disclose the secrets of his employer. Even without a written employment agreement, a confidential relationship between employer and employee is created where the employer discloses to the employee a pre-existing trade secret. This confidential relationship survives even the termination of employment.[44]

Injunctions

Injunction relief - both preliminary and permanent - is available to prevent the misappropriation or misuse of trade secrets, as well as for copyright infringement. The loss of a trade secret is not measurable in terms of money damages. Irreparable harm can be shown in case of the potential loss of an industry leader's present market and the loss of the advantage of being the pioneer in the field and the market leader.[45] The doctrine of laches requires that a request for preliminary injunctive relief be brought as soon as practicable.[46]

> ***"The federal trademark law...is intended to ensure truthfulness in advertising and to eliminate misrepresentations about the inherent quality or characteristics of a product"***

Preemption

Even with a valid trade secret, however, the federal doctrine of 'preemption' may prevent a trade secret owner from asserting misappropriation. Such doctrine requires that any state law claim which could be defined as a federal claim under a federal statute is 'preempted', and invalidated, to the extent that the act complained of is covered by the federal claims. Misappropriation of trade secrets under state law is preempted by copyright claims under federal law where the act complained of is the unauthorised use of a copyrighted work.[47] However, where the misappropriation involves wrongful acquisition or disclosure, then there should be no preemption. In the CA-Altai decision, CA failed to sue the individual who had misappropriated the source code, and that individual's employer (Altai) was not liable for misappropriation which it was not aware of.

Trademark Law

False Description of Origin

The federal trademark law, known as the Lanham Act, is intended to ensure truthfulness in advertising and to eliminate misrepresentations about the inherent quality or characteristics of a product. Section 43(a) of the Lanham Act prohibits the misdescription of the origin of goods or services in interstate or foreign commerce of the United States.[48]

In any infringement case involving alleged misappropriation of trade secret in software, a Lanham Act violation may also exist. To find such a violation, the injured party must prove that the statements were 'patently false, as opposed to merely having a tendency to deceive'.[49] In such a case, it is not necessary to present any proof of actual consumer confusion, but evidence of such confusion would be useful in demonstrating solidly the quantum of damages.[50]

Recent developments in European and European Community advertising law may also render an infringer liable under local laws.[1]

Infringement of Trademark

A federal trademark infringement occurs when anyone uses in US commerce any reproduction, counterfeit, copy, or colorable imitation of a registered mark in connection with the sale, offering for sale, distribution or advertising of any goods or services on or in connection with which such use is likely to cause confusion, or to cause mistake, or to deceive[2].

If a competitor makes an intermediate copy of software so as to achieve interoperability, the competitor is liable for trademark infringement if it fails to remove a false message (by the original author) from the screen display.[3] Thus, even if a developer is entitled to develop an interoperable program, caution must be taken to avoid confusion or misrepresentation.

Trade Dress

At common law, unfair competition occurs when a competitor uses the identifying features of a product's packaging and appearance to confuse the consumer. Legitimate developers should be aware of this cause of action as alternative legal claim.[4]

State Consumer Fraud Statutes

Consumer fraud statutes in many states prevent the use of a fraud in connection with the sale of goods (and sometimes of services) to any 'consumer' in the state. Where a software vendor falsely describes the origin of the software, a prima facie case of consumer probably also occurs.[5]

Conclusion

The *Brown Bag, Computer Associates* and *Lotus v Borland* decisions - all decided in 1992 - will undoubtedly motivate further action at legislative and judicial levels. In the interim, a software developer (as well as his shareholders, investors and users) must carefully consider all forms of protection, both offensive and defensive, in the battle for protection and freedom from infringement of third-party rights. The balance of existing intellectual property protection arrangements should be reviewed in light of these evolving legal principles.

> ***"Important international legal principles may soon conflict with the evolving American appoaches"***

Important international legal principles may soon conflict with the evolving American appoaches. Non-American courts will decide important issues relating to interoperability and the right to study computer software as established by the EC's Software Directive of March 14 1991 and implementing legislation in the EC Member States.

As copyright protection is propounded as the 'preferred' method of protection under the moribund GATT uruguay Round and bilateral international trade agreements, close examination of the limits of copyright law becomes invaluable. Recent American decisions demonstrate the difficulties inherent in a copyright paradigm as such a preferred method for protection of software. Indeed, one appellate court has even suggested that patent law - with its requirements of novelty and non-obviousness as conditions to exclusive rights - is more suitable. As a result, judicial decisions are defining a narrower scope of copyrights in non-literal elements such as user interfaces, macro commands, keystroke sequence and the like.

Clearly, with such an unsettled state of the law, prudent business managers should re-evaluate their intellectual property strategies. Particular attention may be given to the interplay between the different types of intellectual property rights available.

The political balance between protected innovation and unfair monopolistic competition under antitrust law will undoubtedly receive more careful legislative, administrative and judicial attention.

William B. Bierce is a Partner at Bierce & Kenerson, New York City. BA Yale University; Licence en Droit, University of Grenoble, France, J.D., New York University. Former adjunct professor of legal and tax aspects of international business, Pace University Graduate School of Business, New York.

Editors' Note: all American spellings are retained.

Footnotes

[1] *Arrhythmia Research Technology, Inc. v. Corazonix Corp.*, 2d, 1992, US App. LEXIS 4202 (9th Cir. Mar. 12, 1992) (invention includes mathematical algorithm to identify anomalous

electrocardiographic signals of the hearts of patients with ventricular tachycardia).

[2] *Computer Associates International, Inc, v Altai, Inc.*, Civ. No. 91-7893, F.2d, Slip Op. at 42 (2nd Cir. June 22, 1992) [hereinafter, '*Computer Associates*'], affirming 775 F. Supp. 544 (E.D.N.Y. 1991).

[3] 101 US 99 (1879).

[4] 101 US at 104.

[5] This article does not consider the case of *Apple Computer, Inc. v Microsoft Corp.*, F. Supp, 1992 US Dist. LEXIS 5986 (N.D. Calif. 1992 Apr. 15, 1992). That decision was based simultaneously on copyright law and an analysis of the scope of a particular license agreement.

[6] *Whelan Associates, Inc. v Jaslow Dental Lab.*, 797 F.2d 1222 (3rd Cir. 1986), *cert. denied*, 479 US 1031 (1987) [hereinafter, '*Whelan*'].

[7] *Whelan*, 797 F.2d at 1236 (citations omitted).

[8] *Whelan*, 797 F.2d 1222, 1236 (3d Cir. 1986) (scenes a faire, or incidents, characters or settings which are as a practical matter indispensable, are not copyrightable; ract-intensive works are given limited copyright coverage).

[9] *See, eg. Bull HN Information Sys., Inc. v American Express Bank Ltd.*, 1990 Copyright Law Decis (CCH), Para. 26,555, at p. 23,278 (S.D.N.Y. 1990); *Dynamic Solutions, Inc. v Planning & Control, Inc.*, 1987 Copyright Law Decis. (CCH) Para 26,062 at p. 20,912 (S.D.N.Y. 1987); *Digital Communications Associates v Softklone Distributing Corp.*, 659 F. Supp 1127 (N.D. Calif. 1986) (copyright includes audiovisual screen displays); *Broderbund Software Inc. v Unison World, Inc.*, 648 F.Supp 1127 (N.D. Cal. 1986); *SAS Institute, Inc. v S&H Computer Systems, Inc.*, 605 F.Supp. 816 (M.D. Tenn. 1985) (expropriation of design and structure is infringement, even though there is no copying of source code); *cf. Johnson Controls, Inc. v Phoenix Control Systems, Inc.*, 886 F.2d 1173, 1175 (9th Cir. 1989).

[10] *See, eg., Plains Cotton Co-op v Goodpasture Computer Service Inc.*, 807 F.2d 1256, 1262 (5th Cir.) *cert denied*, 484 US 821 (1987) (where market factors played as significant role in determining the sequence and organisation of software, organisational similarity between two programs is not sufficient to support a claim of infringement); *Co-op Industries, Inc. v Hoffman*, 625 F. Supp. 608 (S.D.N.Y. 1985) (where structural similarities between two programs were dictated by functional considerations applicable to the teleprompting industry, there is no infringement). *Cf. Synercom Technology, Inc. v University Computing Co.*, 462 F.Supp 1003, 1014 (N.D. Tex. 1978) (concluding that order and sequence of data or computer input formats in a statistical analysis program are idea and not expression).

[11] See *Shaw v Lindhim*, 919 F.2d 1353 (9th Cir. 1990).

[12] F.2d (9th Cir. Apr. 7, 1992), 1992 U.S. App. LEXIS 6154 [hereinafter, '*Brown Bag*'].

[13] *See, eg., Whelan*, 797 F.2d 1222, 1223, (source code and object code); *CMS Software Design Sys., Inc. v Info Designs. Inc.*, 785 F.2d 1246, 1249 (5th Cir. 1986) (source code); *Apple Computer Inc. v Franklin Computer Corp.*, 714 F.2d 1240, 1249 (3d Cir. 1983), *cert dismissed*, 464 US 1033 (1984) (source and object code); *Williams Electroncis, Inc. v Artic Int'l Inc.*, 685 F.2d 870, 876-877 (3d Cir. 1982) (object code).

[14] *Computer Associates*, Slip Op. at 19.

[15] 17 U.S.C. 102(b); *see Baker v Selden*, 101 US 99 (1879); *Mazer v Stein*, 347 US 201, 217 (1954).

[16] See *Peter Pan Fabrics, Inc v Martin Weiner Corp.*, 274 F.2d 487, 489 (2d Cir. 1960).

[17] Slip Op., pp. 27-28.

[18] *Computer Associates*, slip op., p.29.

[19] Id., p.30.

[20] Slip op., p.48.

[21] Id., pp. 30-32.

[22] Id., p. 33.

[23] See *Brown Bag Software v Symantec Corp.*,.2d,1992 US App. LEXIS 6154, Civ. No. 89-16239, slip op. 3719 (9th Cir. April 7, 1992) (endorsing :analytic dissection' of computer programs to isolate protectable expression, and denying protection to 'commonplace' computing techniques in the software industry).

[24] *Computer Associates*, slip op., pp. 30, 36.

[25] *Feist Publications, Inc. v Rural Telephone Service Co., Inc.*, 111 S. Ct. 1282, 1290 (1990).

[26] *Computer Associates*, slip op., p. 42.

[27] 740 F. Supp 37 (D. Mass 1990)

[28] *Lotus Dev. Corp. v Borland Int'l Inc.*, 788 F.Supp 78 (D. Mass. March 20, 1992); *Lotus Dev. Corp. v Borland Int'l Inc.*, F. Supp., Civ No. 90-11662-K, Slip Op. (D. Mass. July 31, 1992) [hereinafter, '*Lotus*'].

[29] *Lotus* (finding copyright infringement); see also, *Lotus Development Corporation v Borland International, Inc., Development Corporation v Paperback Software International*, 749 F. Supp 37 (D. Mass 1990).

[30] *Lotus*, slip op., pp. 16-17, quoting in part *Lotus Development Corporation v Paperback Software International*, 740 F.Supp. 37, 60-61 (D. Mass. 1990).

[31] Boraland's Mem. in Support of Renewed Motion for Summary Judgment (Docket No. 168), quoted in *Lotus*, slip op. at p.9.

[32] *Lotus*, slip op., p.10.

[33] *Lotus*, slip op., p.15.

[34] O.J.E.C. Mar. 14, 1991).

[35] *Id.*, Art. 5, Para 3.

[36] *Id.*, Art 6.

[37] *Sega Enterprises Ltd v Accolade, Inc.*, 785 F.Supp. 1393, 1396, 1992 US Dist. LEXIS 4028 (N.D. Cal. April 3, 1992) ('If the process of reverse engineering of software entails the duplication of the copyrighted work and the recasting or retransformation of the object code into a form more intelligible to humans, it may infringe upon the copyright owner's exclusive rights').

[38] Restatement of Torts, Section 757, Comment b (1939).

[39] *Eagle Comtronics, Incl. v Pico, Inc.*, 89 A.D. 2d 803, 803-804, 453 N.Y.S.2d 470, 472 (4th Dep't 1982).

[40] *Integrated Cash Management Services Inc, v Digital Transactions, Inc.*, 920 F.2d 171 (2nd Cir. 1990).

[41] 920 F.2d at 174 [citations omitted].

[42] *Computer Associates*, slip op., p.38, citing *Brown Bag*. slip op. 3719, 3738.

[43] *Computer Associates International, Inc. Bryan*, 784 F.Supp 982, 998-999. (E.D.N.Y. Jan. 29, 1992) [hereinafter, '*Bryan*'].

[44] *Healthcare Afiliated Services, Inc. v Lippany*, 701 F.Supp. 1142, 1155 (W.D. Pa. 1988) (applying Pennsylvania law).

[45] *Bryan*, 784 F.Supp. 982, 986.

[46] *See, generally, Citicbank, N.A. v Citytrust*, 756 F.2d 273, 276 (sd Cir. 1985); *Bryan*, 784 F.Supp. 982, 987 (E.D.N.Y. Jan 29, 1992) (a preliminary injunction was granted even after an extensive investigation of 5 months from the discovery of the tort to the request for preliminary injunction, in order to gather the necessary facts required to support a complex action).

[47] *Computer Associates International, Inc v Altal, Inc.*, 775 F.Supp. 544, 564 (E.D.N.Y), affirmed in *Computer Associates.*

[48] Any person who shall affix, apply, or annex, or use in connection with any goods, a false designation of origin, or any false description or representations, including words or other symbols tending falsely to represent the same ... shall be liable to a civil action ... by any person who believes that he is or is likely to be damaged by the use of any such false description or representation. 15 U.S.C. 1125(a).

[49] *Manufacturers Technologies, Inc. v. CAMS, Inc.*, 706 F.Supp. 984, 1003 (D. Conn. 1989).

[50] Id.

[1] See J. Maxeiner and P. Schotthoffer, Advertising Law in Europe and North America (Kluwer 1992).

[2] 15 U.S.C. 1114(1)(a).

[3] *Sega Enterprises Ltd v Accolade, Inc* 862 F.2d 204, 205 (9th Cir 1988).

[4] *See, eg, Data East USA Inc v Epyx Inc*, 862 F.2d 204, 105 (9th Cir 1988).

[5] *Manufacturers Technologies, Inc v Cams, Inc*, 706 F. Supp 984 (D. Conn 1989).