LEGAL PROTECTION OF SOFTWARE: WHY IS MICROSOFT WORTH MORE THAN GENERAL MOTORS?

OREGON STATE BAR COMPUTER AND INTERNET LAW SECTION

January 19, 2000

Timothy E. Siegel
www.intproplaw.com
(503) 973-5984

Users' Concerns and How the Law Addresses a Competitor's Effort to Meet These Concerns:

Similar or Identical Command Set

Reduces cost of training staff (and oneself), which is the greater part of the cost of software.

IP Issues: Is configuration of menus and icons protectable by copyright, or is it an unprotectable "method of operation"? A: There is a split of authority between the circuits on this point.

Can organizational elements be protected by patent? A: A software program cannot display a button for a feature that it cannot offer because of a patent. Software has become increasingly patentable and an increasingly important element in the suite of legal tools used to protect software products.

Present and Future File Reading Compatibility

Will user now and in the future be able to read messages sent to him in industry standard (MSWord) file format?

IP Issues: Can a software purchaser and/or licensee decompile a software program. The better a competitor can understand a software program, the better chance he has of replicating the features that read files. A. Absent a license restriction the user may decompile. The scope of permissibility of license restrictions is not resolved.

Reliability

Will user encounter bugs in software?

IP Issue: Trademark law . Because software cannot be completely evaluated prior to purchase, the use of the MICROSOFT trademark takes on a great importance in assurance of quality to the customer.

Software Features

What array of options does software permit a user to perform? This is probably the least important aspect of the software, given that the product meets some minimum capability.

IP issue: The availability of patent protection for software has been greatly expanded in scope over the last 10 years.

Copyright:

Q.: Is Copyright law an effective tool in combating literal copying. A. Yes, because it presents prospective infringers with a frightening array of remedies, including statutory damages of up to $20,000. 37 USC §504(c). In addition, counterfeiters are subject to criminal prosecution 37 USC §506.

Q. Could someone copy the organization and content of menus and icons without violating Microsoft's Copyright?

A. The First Circuit says yes, the Tenth Circuit says no and the Supreme Court is evenly split.

"We hold that the Lotus menu command hierarchy is an uncopyrightable "**method of operation.**" The Lotus menu command hierarchy provides the means by which users control and operate Lotus 1-2-3. If users wish to copy material, for example, they use the "Copy" command. If users wish to print material, they use the "Print" command. Users must use the command terms to tell the computer what to do. Without the menu command hierarchy, users would not be able to access and control, or indeed make use of, Lotus 1-2-3's functional capabilities." Lotus Development Corp. v. Borland International Inc., 34 USPQ.2d (BNA) 1014, 1021 (1st. Cir. 1995); **Aff'd without opinion by evenly divided Supreme Court 116 S.CT. 804.**

"We conclude that although an element of a work 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. . . . Thus, we decline to adopt the Lotus court's approach to section 102(b), and continue to adhere to our abstraction-filtration-comparison approach. See Gates Rubber , 9 F.3d at 843 (noting the applicability of abstraction- filtration-comparison to "menus and sorting criteria")" Mitel Inc. v. Iqtel Inc., 44 U.S.P.Q.2D (BNA) 1172, 1177 (10th Cir. 1997)

Q. Can Software Originator Prevent Competitor From Decompiling The Machine Code

Background: Computer programs are typically written in a higher-level language ("source code") that is designed to be easily usable by people. The source code is "compiled" to produce "machine code" that is understandable to a computer, but almost impossible for a person to understand. The machine code is typically all that is made available by a software vendor. Having the source code greatly eases the task of a software engineer who is attempting to write a competing program. There are computer programs called "decompilers" that translate backwards, from the machine code to the source code. Typically, software vendors include a license agreement with their software by which the purchaser agrees that he will not decompile the program to obtain the source code.

"You may not reverse engineer, decompile, or disassemble the SOFTWARE PRODUCT, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation." Microsoft End User License Agreement.

A. This depends in large part upon the interpretation of the following Code Section: "It is not an infringement for the owner of a copy of a computer program to make or authorize the making of another copy or adaptation of that computer program provided: (1) that such a new copy or adaptation is created as an essential step in the utilization of the computer program in conjunction with a machine and that it is used in no other manner . . . . 17 U.S.C. Section 117."

No case law directly confronts this issue. The Fifth Circuit has decided that Federal Copyright Law preempted Louisiana from enforcing a law specifically permitting a license restriction against decompilation. Vault Corp. v. Quaid Software Ltd , 7 U.S.P.Q.2D 1281 (5th Cir. 1988) In general, a state law will avoid preemption,. "if an "extra element" is "required instead of or in addition to the acts of reproduction, performance, distribution or display, in order to constitute a state-created cause of action, ..." Computer Assoc. Int'l, Inc. v. Altai, Inc., 982 F.2d 693 (2d Cir. 1992); Accord, National Basketball Association v. Motorola, 41 USPQ 1585 (2d Cir. 1997).

The Ninth Circuit has decided that absent a license provision against decompilation, that decompilation constitutes fair use, in order to gain access to ideas and functional concepts embodied within code. Sega Enterprises Ltd. v. Accolade Inc., 24 U.S.P.Q.2D (BNA) 1561, (9th Cir. 1992). Accord, Alcatel USA Inc. v. DGI Technologies Inc. 49 U.S.P.Q.2D 1641 (5th Cir. 1999)

Based on the above cases, one can surmise that a carefully crafted license clause, forbidding decompilation for the purpose of designing a competing product, would generally be upheld.

An issue raised by 17 U.S.C 117: Can a software purchaser be termed a mere copyright licensee as opposed to owner. A: If the agreement is written correctly, yes. DSC Communications Corp. v. Pulse Communications Inc., 50 U.S.P.Q.2D (BNA) 1001, 1031 (Fed. Cir. 1999)

"All title and intellectual property rights in and to the SOFTWARE PRODUCT (including but not limited to any images, photographs, animations, video, audio, music, text and "applets" incorporated into the SOFTWARE PRODUCT), the accompanying printed materials, and any copies of the SOFTWARE PRODUCT, are owned by MS, or its suppliers (including Microsoft Corporation)" Microsoft End User License Agreement.

Scope of Protection for Nonliteral Copying

Evaluated by abstration-filtration-comparison test. Computer Associates International Inc. v. Altai Inc, **23 U.S.P.Q.2D (BNA) 1241 (2nd Cir. 1992)**

There have not been enough decisions to clearly delineate the scope of protection for nonliteral copying. Here is an interesting few words, however:

"external considerations such as compatibility may negate a finding of infringement." Bateman v. Mnemonics Inc. 38 U.S.P.Q.2D (BNA) 1225 (11th Cir. 1996)

TRADEMARK LAW

Q. Could I Market SIEGELWORD?

A. Microsoft does not appear to own a Federal Registration for WORD. (It does own MICROSOFT POCKET WORD) Nevertheless, it is likely that Microsoft has built up common law rights in this mark and could mount a well financed suit against me if I started using the mark SIEGELWORD for my look alike word processing program.

PATENT LAW

The patentability of software has grown considerably in the last ten years. In State Street Bank & Trust Co., v. Signature Financial Group, Inc. 149 F.3d 1368; 47 U.S.P.Q.2D 1596 (Fed. Cir. 1998) the bar against patenting business methods was removed. In addition the physical transformation that must be mandated by an algorithm was defined so broadly, that virtually any program step would satisfy this requirement. Accord, AT & T Corp. v. Excel Communications Inc., 50 U.S.P.Q.2D 1447 (Fed. Cir. 1999).

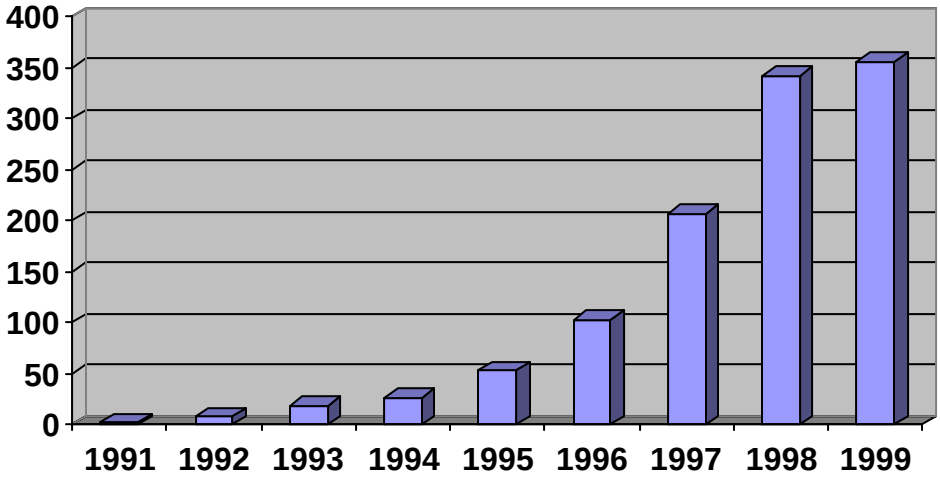
Protects program features, which also are typically user interface features. So, competitor cannot present look alike program.

There follow a few patent examples of patent claims that are owned by Microsoft and protect various word processing features.

The first example covers the feature in which MSWord figures out how you are formatting your paragraphs and starts doing it for you. Although I personally find this feature infuriating, it is an interesting example of what can be covered in a software patent claim.

The second example covers a feature that lets a user replace, for example, all the conjugations of the word “purchase” with the word “buy” so that “purchased” would be replaced by “bought” rather than “buyed” as it would be for a string search and replace.

Number of Patents Issued to Microsoft as Assignee



United States Patent **5,900,004**
Gipson **May 4, 1999**

Method and system for interactive formatting of word processing documents with deferred rule evaluation and format editing

1. A method for interactively formatting a word processing document in a word processor system including a programmed computer, a display device, and an input device, the method comprising the steps of:

(a) intercepting user input events received from the input device as a user enters input events while preparing a word processing document;

(b) analyzing the user input events between user input events and without explicit initiation of the analyzing step by the user to determine whether the user input events correspond to predefined events, where the predefined events are arguments to expressions of autoformat rules, and the expressions are conditions that need to be satisfied for the autoformat rules to generate autoformat actions;

(c) scheduling a rule for evaluation by queuing the rule for evaluation in response to detecting that the at least one predefined event has occurred and is an argument to the rule;

(d) evaluating an autoformat rule between user input events and without explicit initiation of the evaluating step by the user in response to detecting that at least one of the predefined events has occurred; and

(e) in response to evaluating the rule, performing an autoformat action corresponding to the rule automatically between user input events as the user enters the input events while preparing the word processing document to change formatting of the document.

United States Patent **5,873,660**
Walsh , et al. **February 23, 1999**

Morphological search and ***replace***

1. In a word processor for a computer system, a method of finding and replacing forms of a find word with forms of a replace word in a text document, the method comprising:

storing a plurality of word form sets in a word forms database, each word form set comprising a plurality of word forms having a same word base, each word form being associated with at least one part-of-speech identifier;

searching the word forms database for any word form sets that contain a word form matching the find word, and for any word form sets that contain a word form matching the replace word;

in a case where multiple word form sets each contain a word form matching the find word, selecting a single find word form set out of the multiple word form sets that contain a word form matching the find word;

in a case where multiple word form sets each contain a word form matching the replace word form set out of the multiple word form sets that contain a word form matching the replace word;

searching the text document for a target word that matches any word form in the find word form set, the matched word form in the find word form set being a found word form;

searching the replace word form set for a replacement word form whose associated part-of-speech identifier matches a part-of-speech identifier associated with the found word form; and

replacing the target word with the replacement word form.