Against User Interface Copyright

(August 26, 1991)

The League for Programming Freedom

In June 1990, Lotus won a copyright infringement suit against Paperback Software, a small company that implemented a spreadsheet that obeys the same keystroke commands used in Lotus 1-2-3. Paperback was not accused of copying code from 1-2-3—only of supporting compatible user commands. Such imitation was common practice until unexpected court decisions in recent years extended the scope of copyright law.

Within a week, Lotus went on to sue Borland over Quattro, a spreadsheet whose usual command language has only a few similarities to 1-2-3. Lotus claims that these similarities in keystroke sequences and/or the ability to customize the interface to emulate 1-2-3 are enough to infringe.

More ominously, Apple Computer has sued Microsoft and Hewlett Packard for implementing a window system whose displays partially resemble those of the Macintosh system. Subsequently Xerox sued Apple for implementing the Macintosh system, which derives some general concepts from the earlier Xerox Star system. These suits try to broaden the Lotus decision and establish copyright on a large class of user interfaces. The Xerox lawsuit was dismissed because of a technicality; but if it had succeeded, it would probably have created an even broader monopoly than the Apple lawsuit may.

And Ashton-Tate has sued Fox Software for implementing a database program that accepts the same programming language used in dBase. This is a radical demand, but in the current judicial climate, the threat cannot be neglected.[1] In the same vein, Adobe claims that the Postscript language is copyrighted, though it has not sued those who reject this claim. If a programming language becomes copyrighted, the impact on users who have spent years writing programs in the language would be devastating.

While this paper addresses primarily the issue of copyright on specific user interfaces, most of the arguments apply with added force to any broader monopoly.

What Is a User Interface?

A user interface is what you have to learn to operate a machine; in other words, it is the language you use to communicate with the machine. The user interface of a typewriter is the layout of the keys. The user interface of a car includes a steering wheel for turning, pedals to speed up and slow down, a lever to signal turns, etc.

When the machine is a computer program, the interface includes that of the computer—its keyboard, screen and mouse—plus those aspects specific to the program. These typically include the commands, menus, programming languages, and the way data is presented on the screen.

A copyright on a user interface means a government-imposed monopoly on its use. In the example of the typewriter, this would mean that each manufacturer would be forced to arrange the keys in a different layout.

The Purpose of Copyright

In the United States, the Constitution says that the purpose of copyright is to "promote the progress of science and the useful arts." Conspicuously absent is any hint of intention to enrich copyright holders to the detriment of the users of copyrighted works.

---

[1] Borland bought Aston-Tate in July 1991, thus placing itself in the ironic position of being a defendant in one interface copyright lawsuit and a plaintiff in another. It remains to be seen whether Borland will drop the Ashton-Tate lawsuit.

The Supreme Court made the reason for this absence explicit, stating in *Fox Film vs. Doyal* that "The sole interest of the United States and the primary object in conferring the [copyright] monopoly lie in the general benefits derived by the public from the labors of authors."

In other words, since copyright is a government-imposed monopoly, which interferes with the freedom of the public in a significant way, it is justified only if the benefit to the public exceeds the cost to the public.

The spirit of individual freedom must, if anything, incline us against monopoly. Following either the Supreme Court or the principle of freedom, the fundamental question is: what value does user interface copyright offer the public—and what price would we have to pay for it?

Reason #1: More Incentive Is Not Needed

The developers of the Star, the Macintosh system, 1-2-3 and dBase claim that without interface copyright there would be insufficient incentive to develop such products. This is disproved by their own actions.

Until 1986, user interface copyright was unheard of. The computer industry developed under a system where imitating a user interface was both standard practice and lawful. Under this system, today's plaintiffs made their decisions to develop their products. When faced with the choice in actuality, they decided that they did, indeed, have "enough incentive."

Even though competitors were free to imitate these interfaces, this did not prevent most of the original products from being successful and producing a large return on the investment. In fact, they were so successful that they became *de facto* standards. (The Xerox Star was a failure due to poor marketing even though nothing similar existed.)

Even if interface copyright would increase the existing incentive, additional improvements in user interfaces would not necessarily result. Once you suck a bottle dry, more suction won't get more out of it. The existing incentive is so great that it may well suffice to motivate everyone who has an idea worth developing. Extra incentive, at the public's expense, will only increase the price of these developments.

Reason #2: "Look and Feel" Will Not Protect Small Companies

The proponents of user interface copyright claim that it would protect small companies from being wiped out by large competitors. Yet look around: today's interface copyright plaintiffs are large, established companies. User interface copyright is crushing when the interface is an effective standard. However, a small company is vulnerable when its product is little used, and its interface is little known. In this situation, user interface copyright won't help the small company much.

Imagine a small company with 10,000 customers: a large company may believe there is a potential market of a million users, not reached by the small company, for a similar product. The large company will try to use its marketing might to reach them before the small company can.

User interface copyright won't change this outcome. Forcing the large company to develop an incompatible interface will have little effect on the majority of potential customers—those who have not learned the other interface. They will buy from the large company anyway.

What's more, interface copyright will work against the small company if the large company's product becomes an effective standard. Then new customers will have an additional reason to prefer the large company. To survive, the small company will need to offer compatibility with this standard—but, due to user interface copyright, it will not be allowed to do so.

Instead of relying upon monopolistic measures, small companies are most successful when they rely on their own inherent advantages: agility, low overhead, and willingness to take risks.

Reason #3: Diversity in Interfaces Is Not Desirable

The copyright system was designed to encourage diversity; its details work toward this end. Diversity is the primary goal when it comes to novels, songs, and the other traditional domains of copyright. Readers want to read novels they have not yet read.

But diversity is not the goal of interface design. Users of any kind of machinery want consistency in inter-

faces because this promotes ease of use. Thus, by standardizing symbols on automobile dashboards, we have made it possible for any licensed driver to operate any car without additional instruction. Incompatibility in interfaces is a price to be paid when worthwhile, not a benefit.

Significantly better interfaces may be hard to think of, but it is easy to invent interfaces which are merely different. Interface copyright will surely succeed in encouraging this sort of "interface development." The result will be gratuitous incompatibility.

Reason #4: Meaningful Competition Is Reduced

Under the regime of interface copyright, there will be no compatible competition for established products. For a user to switch to a different brand will require retraining.

But users don't like to retrain, not even for a significant improvement. For example, the Dvorak keyboard layout, invented several decades ago, enables a typist to type faster and more accurately than is possible with the standard "QWERTY" layout. Nonetheless, few people use it. Even new typists don't learn Dvorak, because they want to learn the layout used on most typewriters.

Alternative products that require such an effort by the consumer are not effective competition. The monopoly on the established interface will yield in practice a monopoly on the functionality accessed by it. This will cause higher prices and less technological advancement—a windfall for lucky businesses, but bad for the public at large.

Reason #5: Incompatibility Does Not Go Away

If there had been a 50-year interface copyright for the steering wheel, it would have expired not long ago. During the span of the copyright, we would have got cars steered with joysticks, cars steered with levers, and cars steered with pedals. Each car user would have had to choose a brand of car to learn to drive, and it would not be easy to switch.

The expiration of the copyright would have freed manufacturers to switch to the best of the known interfaces. But if Ford cars were steered with wheels and General Motors were steered with pedals, neither company could change interface without abandoning their old customers. It would take decades to converge on a single interface.

Reason #6: Users Invest More Than Developers

The plaintiffs like to claim that user interfaces represent large investments on their part.

In fact, the effort spent designing the user interface of a computer program is usually small compared to the cost of developing the program itself. The people who make a large investment in the user interface are the users who train to use it. Users have spent much more time and money learning to use 1-2-3 than Lotus spent developing the entire program, let alone what Lotus spent develop the program's interface *per se*.

Thus, if investment justifies ownership, it is the users who should be the owners. The users should be allowed to decide—in the marketplace—who may use it. According to *Infoworld* (mid January 1989), computer users in general expect user interface copyright to be harmful.

Reason #7: Discrimination Against Software Sharing

User interface copyright discriminates against freely redistributable software, such as freeware, shareware and public domain software.

Although it *may* be possible to license an interface for a proprietary program, if the owner is willing, these licenses require payment, usually per copy. There is no way to collect this payment for a freely redistributable program. The result will be a growing body of interfaces that are barred to non-proprietary software.

Authors of these programs donate to the public the right to share them, and sometimes also to study and change their workings. This is a public service, and one less common than innovation. It does not make sense to encourage innovation of one sort with means that bar donation of another sort.

Reason #8: Copyright Will Be a Tool For Extortion

The scope of interface copyright is so vague and potentially wide that it will be difficult for any programmer to be sure of being safe from lawsuits. Most programs need an interface, and there is usually no way to design an interface except based on the ideas you have seen used elsewhere. Only a great genius would be likely to envision a usable interface without a deep resemblance to current practice. It follows that most programming projects will risk an interface infringement suit.

The spirit of "Millions for defense, but not a cent for tribute" is little honored in business today. Customers and investors often avoid companies that are targets of suits; an eventual victory may come years too late to prevent great loss or even bankruptcy. Therefore, when offered a choice between paying royalties and being sued, most businesses pay, even if they would probably win a suit.

Since this tendency is well known, companies often take advantage of it by filing or threatening suits they are unlikely to win. As long as any interface copyright exists, this form of extortion will broaden its effective scope.

Reason #9: Useful Innovation Is Inhibited

Due to the evolutionary nature of interface development, interface copyright will actually retard progress.

Fully fleshed-out interfaces don't often arise as *tours de force* from the minds of isolated masters. They result from repeated implementations, by different groups, each learning from the results of previous attempts. For example, the Macintosh interface was based on ideas tried previously by Xerox and SRI, and before that by the Stanford Artificial Intelligence Laboratory. The Xerox Star also drew on the interface ideas that came from SRI and SAIL. 1-2-3 adapted the interface ideas of Visicalc and other spreadsheets. dBase drew on a program developed at the Jet Propulsion Laboratory.

This evolutionary process resembles the creation of folk art rather than the way symphonies, novels or films are made. The advances that we ought to encourage are most often small, localized changes to what someone else has done. If each interface has an owner, it will be difficult to implement such ideas. Even assuming the owner will license the interface that is to be improved, the inconvenience and expense would discourage all but the most determined.

Users often appreciate small, incremental changes that make programs easier or faster to use. This means changes that are upwards compatible, or affect only part of a well-known interface. Thus, on computer keyboards, we now have function keys, arrow keys, a delete key and a control key, which typewriters did not have. But the layout of the letters is unchanged.

However, such partial changes as this are not permitted by copyright law. If any significant portion of the new interface is the same as a copyrighted interface, the new interface is illegal.

Reason #10: Interface Developers Don't Want Interface Copyright

At the 1989 ACM Conference on Computer-Human Interaction, Professor Samuelson of the Emory School of Law presented a "mock trial" with legal arguments for and against user interface copyright, and then asked the attendees—researchers and developers of user interfaces—to fill out a survey of their opinion on the subject.

The respondents overwhelmingly opposed all aspects of user interface copyright, by as much as 4 to 1 for some aspects. When they were asked whether user interface copyright would harm or help the field, on a scale from 1 (harm) to 5 (help), the average answer was 1.6.[2]

The advocates of user interface copyright say that it would provide better security and income for user interface designers. However, the survey shows that these supposed beneficiaries would prefer to be let alone.

---

[2]See the May 1990 issue of the *Communications of the ACM*, for the full results.

Do You Really Want a User Interface Copyright?

For a business, "locking in" customers may be profitable for a time. But, as the vendors of proprietary operating systems have found out, this generates resentment and eventually drives customers to try to escape. In the long run, this leads to failure.

Therefore, by permitting user interface copyright, society encourages counterproductive thinking in its businesses. Not all businesses can resist this temptation; let us not tempt them.

Conclusion

Monopolies on user interfaces do not serve the users and do not "promote the progress of science and the useful arts." User interfaces ought to be the common property of all, as they undisputedly were until a few years ago.

What You Can Do

- Don't do business as usual with the plaintiffs, Xerox, Lotus, Apple and Ashton-Tate. Buy from their competitors instead; sell their stock; develop new software for other computer systems rather than theirs, and port existing applications away from their systems.

- Don't work for the "look and feel" plaintiffs or accept contracts from them.

- Join the League for Programming Freedom—a grass-roots organization of programmers and users opposing software patents and interface copyrights. (The League is not opposed to copyright on individual programs.) Annual dues are $42 for employed professionals, $10.50 for students, and $21 for others. We appreciate activists, but members who cannot contribute their time are also welcome.

  Phone us at (617) 243-4091, send Internet mail to **`league@prep.ai.mit.edu`**, or write to:

  League for Programming Freedom
  1 Kendall Square #143
  P.O. Box 9171
  Cambridge, MA 02139

- Give copies of this paper to your friends, colleagues and customers.

- In the United States, write to your representatives and to these Congressional subcommittees:

  House Subcommittee on Intellectual Property
  2137 Rayburn Bldg
  Washington, DC 20515

  Senate Subcommittee on Patents, Trademarks and Copyrights
  United States Senate
  Washington, DC 20510

- The European Community has adopted a directive whose most natural interpretation imposes copyright on all kinds of interfaces, even on programming languages. Since the other countries of Europe are considering joining the EC, they also are in danger of being covered by the directive.

  Other, benign interpretations of the directive are also possible, but they are unlikely to be chosen by judges unless the governments of the individual EC countries explicitly mandate them. Convincing the governments requires political pressure from the programmers and users of Europe.

Lobbyists working on this issue say that most legislators are unfamiliar with computers and do not understand how harmful interface copyright could be. Thus, what programmers need to do is to educate their legislators.

One idea is to start teaching your representative the basics of using 1-2-3. Once the representative sees how much work is involved in learning to use a command language, explain that you have only taught one tenth of the subject. This should drive the point home.

Political effectiveness requires organization. Leagues for Programming Freedom now exist in Finland, Germany, the United Kingdom, the Netherlands, Norway, and Switzerland. (In the UK, the Edinburgh Computing and Social Responsibility organization also deals with this issue.) Ask the League in the US for the address of your nation's League—or for advice and assistance in forming one.