Actually, it must be conceded that the idea/expression distinction has served us fairly well in a multitude of cases involving books, plays, and other traditional subject matter of copyright. But it does not help us determine whether it is a good or bad notion to protect computer program user-interfaces, icons, metaphors, menus, screens, instruction sets, languages, algorithms, flowcharts, dataflow patterns, and other noncode computer-program paraphernalia in some way and to some extent from unlimited competitive appropriation.

The distinction is also unhelpful when trying to decide whether similarly to protect "blown" PALS.

In the first place, it may be possible to distinguish idea and expression in the case of poems, novels, and plays, and perhaps even to understand what we mean by those terms in a literary or artistic context. But it is by no means practical to distinguish idea and expression in computer programs, and it is probably extremely optimistic even to think that we know what the terms mean when applied to computer software-related subject matter (what I have termed computer-program paraphernalia).

In the second place, even if we knew what we were talking about and could sensibly distinguish idea from expression, it would not be particularly useful in the computer-program context. Why should ideas not be protected in some way for computer-program paraphernalia? What is so terrific about protecting expressions and not protecting ideas?

Probably there are two general, theoretical copyright law reasons, and no more; neither one of them is relevant in the computer-program context. The first reason concerns free speech. If ideas can be monopolized by copyright, others' speech and self-expression will be hindered. But we would rather that they be given enough breathing room to foster the kind of open society that we prize (where one can mouth off to at least a certain extent). Second, there is the economic rationale described above: that we do not want to hand out what amounts to patents without first having a patent office evaluate the new subject matter.

The first factor definitely is out of place for computer-program paraphernalia. Writing a computer program is not the kind of self-expression that the First Amendment is all about. Anyway, why (for one example) would paying Nicolas Wirth a small royalty fee for the benefit of using his Modula-2 instruction set or language be such a hindrance to self-expression? The same might even be said for the creator of the desktop metaphor or the garbage-can icon. After all, I did not suggest injunctions or throwing people in jail for using somebody's software paraphernalia. The creative incentive or investment incentive of the royalty may well outweigh any hindrance to progress or self-expression.

---

**Myths block sensible solutions: They cause the wrong questions to be asked and block the right questions.**

The second factor can in principle be dealt with in several ways. One is to forget about copyright law's automatic protection of anything independently created, no matter how inferior, and instead examine the technological merit of the instruction set, icon, etc., as under a patent system. That could be done before according any economic rights. (I believe that advance examination of merit would be a bad idea, however, because such a system cannot handle too much of a front-end load.) Or we could (preferably) examine technological merit at the time of a dispute, if one ever occurs. Finally, the equilibrium of incentive/hindrance trade-offs can be modulated by providing only modest economic rights (a small royalty to Wirth, a mere pittance to the garbage-can-icon person, in my previous examples).

I plan to return to the foregoing software-paraphernalia protection scheme in a future column. But at this time, the point I want to make is only that taking a legal myth too seriously may prevent one from fashioning a sound solution to the further ramifications of the problems that led to the fabrication of the myth. Here, the myth starts out because of a feeling, "Gee whiz, we ought to do something for software proprietors," which is followed up by fabrication of the myth that a computer program is a book and thus protectable under copyright law. That leads to all the other legal baggage and impedimenta of book copyrights, such as expressions are protectable and ideas are unprotectable. That then leads to total paralysis or confusion when it turns out that the most economically valuable aspects of software seem to fall into the idea category. The myth has completely blocked sensible solution of the problem. It causes the wrong questions to be asked, questions whose answers are of no practical use; and it blocks our asking the right questions, whose answers might help us frame a sensible solution to problems.

The myth that a program is a book ends up by helping software publishers suppress pirates of mass-marketed software (such as word processor programs and games), and perhaps by helping mainframe manufacturers keep emulator manufacturers from using the operating system under which the installed base of applications software runs, thereby preventing customers from being diverted to the emulators.

Doubtless both of these are worthwhile and socially beneficial projects. But the same myth keeps us from rewarding the creators of new instruction sets, languages, algorithms, and the other valuable ideas that fuel software progress. It even stands in the way of giving file-deletion icon creators their just rewards.

Could you say that about the mythology of IC design?

---

**Reader Interest Survey**

Indicate your interest in this department by circling the appropriate number on the Reader Interest Card.

High 177  Medium 178  Low 179