

Intellectual Property

Adam Goode
Computer Ethics

December 6, 2000

Contents

1 Introduction	1
2 Traditional Ideas About Property	1
3 Ideas as Property	2
4 Traditional Property Notions Applied To Intellectual Items	3
5 Software Should Be Free	5
6 Conclusion	5

1 Introduction

Intellectual property is a controversial topic. The term itself is often ill-defined and confusing. According to one definition, intellectual property is “The ownership of ideas and control over the tangible or virtual representation of those ideas (Howe, 2000).” But can ideas really have owners and control? Some (notably those like Richard Stallman) would advocate “no.” And of course, most software business leaders would give a powerful “yes.”

The author of this paper is under the persuasion that intellectual property does not make sense (at least in software) and that control over one’s ideas is something that one gives up the moment he releases them to the world. This notion will be examined, as well as the notions of traditional property, ideas as property, where intellectual property seems to fit in with software, and why it really cannot.

2 Traditional Ideas About Property

The word “property” brings out certain specific mental images. Ideas that spring to mind with the mention of property include things like “house” (dwelling as

property), “car” (machine as property), “yard” (land as property), or even “pet” (animal as property). Not many people initially think of “idea” or “notion” as a thing which might have property.

Property has generally been restricted to things in the physical realm. Certainly a person’s house or car can be considered property. Both are physical, and both have the unique attribute that they exist only once. Some explanation on that point: Sure it’s possible to have more than one car, even more than one car of a particular model. But each car is individual, and it would be wrong to look at two similar cars and claim that they are the same thing. This notion of *numerical identity* implies a type of conservation of mass: a single car is a single car, and if an owner parts with that particular car, that owner loses his rights to it as the new owner gains rights. You can’t make two cars from one: one man’s gain is another’s loss.

Deborah Johnson gives her own views on traditional property:

Property is by no means a simple notion. Property is created by law. Laws specify what can and cannot be owned, how things may be acquired and transferred. Laws define what counts as property and create different kinds of property. The laws regulating ownership of an automobile, for example, are quite different from those regulating ownership of land. In the case of land, there are rules about how far the land goes down into the ground and how far up into the air space above, about what can and cannot be constructed on the land, when the land may be confiscated by the government, and so on. With automobiles, the laws are quite different. You may have to show proof of insurance in order to buy a car, and even if you own one, you cannot drive it on public roads unless you have a license. Thus, how you come to own something and what it means to own something are rather complex matters (Johnson, 1994, p. 70).

It’s clear that property is a tricky, complex issue. But it just gets worse.

3 Ideas as Property

Recently (within the past century or so), people began to see *ideas* as having some of the same attributes as physical property had. “Ideas have to come from somewhere—and yes, they come from people, just like other inventions and devices. And because ideas are created,” the argument goes, “the creator should have control over what he produces. After all, it would be wrong for another man to take my house away from me. What gives others the right to take my ideas?”

At first glance, this argument may seem to be sound. Unfortunately, the argument makes at least one fatal assumption: ideas are not physical items. When a person invents an idea, he can share it with everyone without loss or degradation of the original. Try that with your house or car. Ideas, because they are very different from physical objects, can be shared without any loss.

At this point, it is unclear how to handle ideas as a form of property. It may not even make sense. But some people have tried. Indeed, the international software industry bases their entire business on this premise: that ideas are property. We will look at this next.

4 Traditional Property Notions Applied To Intellectual Items

Here is a letter from the dawn of the modern age of computing, quoted in its entirety:

An Open Letter to Hobbyists
February 3, 1976
By William Henry Gates III

To me, the most critical thing in the hobby market right now is the lack of good software courses, books and software itself. Without good software and an owner who understands programming, a hobby computer is wasted. Will quality software be written for the hobby market?

Almost a year ago, Paul Allen and myself, expecting the hobby market to expand, hired Monte Davidoff and developed Altair BASIC. Though the initial work took only two months, the three of us have spent most of the last year documenting, improving and adding features to BASIC. Now we have 4K, 8K, EXTENDED, ROM and DISK BASIC. The value of the computer time we have used exceeds \$40,000.

The feedback we have gotten from the hundreds of people who say they are using BASIC has all been positive. Two surprising things are apparent, however, 1) Most of these "users" never bought BASIC (less than 10% of all Altair owners have bought BASIC), and 2) The amount of royalties we have received from sales to hobbyists makes the time spent on Altair BASIC worth less than \$2 an hour.

Why is this? As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid?

Is this fair? One thing you don't do by stealing software is get back at MITS for some problem you may have had. MITS doesn't make money selling software. The royalty paid to us, the manual, the tape and the overhead make it a break-even operation. One thing you do do is prevent good software from being written. Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product

and distribute for free? The fact is, no one besides us has invested a lot of money in hobby software. We have written 6800 BASIC, and are writing 8080 APL and 6800 APL, but there is very little incentive to make this software available to hobbyists. Most directly, the thing you do is theft.

What about the guys who re-sell Altair BASIC, aren't they making money on hobby software? Yes, but those who have been reported to us may lose in the end. They are the ones who give hobbyists a bad name, and should be kicked out of any club meeting they show up at.

I would appreciate letters from any one who wants to pay up, or has a suggestion or comment. Just write to me at 1180 Alvarado SE, #114, Albuquerque, New Mexico, 87108. Nothing would please me more than being able to hire ten programmers and deluge the hobby market with good software.

Bill Gates
General Partner, Micro-Soft
(Gates, 1976)

This letter from Mr. Gates is one of the earliest attacks on the free copying of software. It puts forth Gates's assertion that, as a software designer, that which he has created belongs only to him, and those that use his software (or ideas) may do so only under his terms.

Note his use of the word "theft." Gates is trying to map the ideas associated with physical items onto non-physical items. He is attempting to assert control over his ideas in the same way that he is allowed to assert his control over his car or his dwelling.

Gates's motivations are clear: controlling an idea gives him the ability to control people who employ that idea. He then gains power, and potentially wealth. Indeed, Bill Gates was successful. He is now the richest man in the world, thanks largely to his ideas embodied in this letter. We now have a huge industry built up, and a great economy. We also have a new term to describe this intellectual "theft": piracy.

According to the Software & Information Industry Association,

...the problem of software theft has developed, and threatens to impede the development of new software products. Romantically called "piracy," the unauthorized duplication of software is a Federal offense that affects everyone: large and small software publishers and legitimate users. Even the users of unlawful copies suffer from their own illegal actions. They receive no documentation, no customer support and no information about product updates (SIIA, 2000).

This issue of piracy is made to sound serious. Indeed, the next section of the document is entitled: “When a Few People Steal Software, Everyone Loses.” Of course, this is a natural stance from an organization designed to protect the wealth of large software companies. A huge portion of the assets of a company like Microsoft is based in intellectual property. Should the validity of IP disappear, the software giants would have a lot to lose.

5 Software Should Be Free

Richard M. Stallman says that software should be free. (That’s free as in *freedom*, not necessarily free as in price.) His attacks on proprietary software are well-known and well debated.

His original argument against proprietary software includes the debunking of two main arguments used by software companies: the emotional argument and the economic argument. Stallman begins: “The emotional argument goes like this: ‘I put my sweat, my heart, my soul into this program. It comes from me, it’s mine!’” Stallman goes onto refute the argument, giving the reason that “the feeling of attachment is one that programmers can cultivate when it suits them; it is not inevitable. Consider, for example, how willingly the same programmers usually sign over all rights to a large corporation for a salary; the emotional attachment mysteriously vanishes (Stallman, 1992).”

As for the economic argument, Stallman gives this as the description: “I want to get rich (usually described inaccurately as ‘making a living’), and if you don’t allow me to get rich by programming, then I won’t program. Everyone else is like me, so nobody will ever program. And then you’ll be stuck with no programs at all! (Stallman, 1992)” He spends the rest of the paper debunking this and other arguments, mostly exploring the notion of social harm caused by proprietary software.

6 Conclusion

Looking at the arguments of software industry versus people like Richard Stallman, it becomes clear that Stallman has the correct idea. Intellectual property does not make very good sense, because ideas aren’t physical objects, and thus they can be shared without cost. People shouldn’t be given the right to restrict another’s rights through the use of licensing. People should have the ability to use ideas unrestricted, because ideas are fundamentally free. Once an idea leaves a person, it belongs to everyone. As Thomas Jefferson said,

He who receives an idea from me, receives instruction himself without lessening mine; as he who lights his taper at mine, receives light without darkening me.

References

Gates, III, W. H. (1976). An open letter to hobbyists. Letter to Homebrew Computer Club.

Howe, D., editor (2000). *The Free On-line Dictionary of Computing*. Imperial College Department of Computing, <http://www.foldoc.org/>.

Johnson, D. G. (1994). *Computer Ethics*. Prentice Hall, second edition.

SIIA (2000). Copyright law & related issues: Software use and the law. <http://www.spa.org/piracy/copyright/law.asp>.

Stallman, R. (1992). Why software should be free. <http://www.gnu.org/philosophy/shouldbefree.html>.