

THE NETWORK OBSERVER

The Network Observer (TNO) is a free on-line newsletter about networks and democracy edited by Phil Agre. It appeared monthly from January 1994 to July 1996.

TNO 1(5), May 1994.

- How to help people use computers
- The unfortunate "frontier" metaphor
- E-mail and global non-profit organizations
- Crusade against misleading bouncemail
- New tracking technologies

How to help someone use a computer.

Computer people are generally fine human beings, but nonetheless they do a lot of inadvertent harm in the ways they "help" other people with their computer problems. Now that we're trying to get everyone on the net, I thought it might be helpful to write down in one place everything I've been taught about how to help people use computers.

First you have to tell yourself some things:

- * Nobody is born knowing this stuff.
- * You've forgotten what it's like to be a beginner.
- * If it's not obvious to them, it's not obvious.
- * A computer is a means to an end. The person you're helping probably cares mostly about the end. This is reasonable.
- * Their knowledge of the computer is grounded in what they can do and see -- "when I do this, it does that". They need to develop a deeper understanding, of course, but this can only happen slowly, and not through abstract theory but through the real, concrete situations they encounter in their work.
- * By the time they ask you for help, they've probably tried several different things. As a result, their computer might be in a strange state. That's not their fault.
- * The best way to learn is through apprenticeship -- that is, by doing some real task together with someone who has skills that you don't have.

* Your primary goal is not to solve their problem. Your primary goal is to help them become one notch more capable of solving their problem on their own. So it's okay if they take notes.

* Most user interfaces are terrible. When people make mistakes it's usually the fault of the interface. You've forgotten how many ways you've learned to adapt to bad interfaces. You've forgotten how many things you once assumed that the interface would be able to do for you.

* Knowledge lives in communities, not individuals. A computer user who's not part of a community of computer users is going to have a harder time of it than one who is.

Having convinced yourself of these things, you are more likely to follow some important rules:

* Don't take the keyboard. Let them do all the typing, even if it's slower that way, and even if you have to point them to each and every key they need to type. That's the only way they're going to learn from the interaction.

* Find out what they're really trying to do. Is there another way to go about it?

* Attend to the symbolism of the interaction. Most especially, try not to tower over them. If at all possible, squat down so your eyes are just below the level of theirs. When they're looking at the computer, look at the computer. When they're looking at you, look back at them.

* If something is true, show them how they can see it's true.

* Be aware of how abstract your language is. For example, "Get into the editor" is abstract and "press this key" is concrete. Don't say anything unless you intend for them to understand it. Keep adjusting your language downward towards concrete units until they start to get it, then slowly adjust back up towards greater abstraction so long as they're following you. When formulating a take-home lesson ("when it does this and that, you should check such-and-such"), check once again that you're using language of the right degree of abstraction for this user right now.

* Whenever they start to blame themselves, blame the computer, no matter how many times it takes, in a calm, authoritative tone of voice. When they get nailed by a false assumption about the computer's behavior, tell them their assumption was reasonable. Tell *yourself* that it was reasonable. It was.

* Never do something for someone that they are capable of doing for themselves.

* Don't say "it's in the manual". (You probably knew that.)