How to help someone use a computer.

by Phil Agre

Computer people are fine human beings, but they do a lot of harm in the ways they "help" other people with their computer problems. Now that we're trying to get everyone online, I thought it might be helpful to write down everything I've been taught about helping 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, but this can only happen slowly -- and not through abstract theory but through the real, concrete situations they encounter in their work.
- Beginners face a language problem: they can't ask questions because they don't know what the words mean, they can't know what the words mean until they can successfully use the system, and they can't successfully use the system because they can't ask questions.
- You are the voice of authority. Your words can wound.
- Computers often present their users with textual messages, but the users often don't read them.
- By the time they ask you for help, they've probably tried several things. As a result, their computer might be in a strange state. This is natural.
- They might be afraid that you're going to blame them for the problem.
- The best way to learn is through apprenticeship -- that is, by doing some real task together with someone who has a different set of skills.
- 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.
- Knowledge lives in communities, not individuals. A computer user who's part of a community of computer users will have an easier time than one who isn't.

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 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?

- Maybe they can't tell you what they've done or what happened. In this case you can ask them what they are trying to do and say, "Show me how you do that".

- Attend to the symbolism of the interaction. Try to squat down so your eyes are just below the level of theirs. When they look at the computer, look at the computer. When they look at you, look back at them.

- When they do something wrong, don't say "no" or "that's wrong". They'll often respond by doing something else that's wrong. Instead, just tell them what to do and why.

- Try not to ask yes-or-no questions. Nobody wants to look foolish, so their answer is likely to be a guess. "Did you attach to the file server?" will get you less information than "What did you do after you turned the computer on?".

- Explain your thinking. Don't make it mysterious. If something is true, show them how they can see it's true. When you don't know, say "I don't know". When you're guessing, say "let's try ... because ...". Resist the temptation to appear all-knowing. Help them learn to think the problem through.

- Be aware of how abstract your language is. "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 try such-and-such"), check once again that you're using language of the right degree of abstraction for this user right now.

- Tell them to really read the messages, such as errors, that the computer generates.

- Whenever they start to blame themselves, respond by blaming the computer. Then keep on blaming the computer, no matter how many times it takes, in a calm, authoritative tone of voice. If you need to show off, show off your ability to criticize bad design. 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.

- Take a long-term view. Who do users in this community get help from? If you focus on building that person's skills, the skills will diffuse to everyone else.

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

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

This project is made possible by a grant from the U.S. Institute of Museum and Library Services and the Nebraska Library Commission.