COMPUTERS TALKING TO PEOPLE

It's interesting that with all this preoccupation with getting instructions into the operating system, there is comparatively little obvious concern with getting information out. On may systems it can be quite hard to be sure that the instruction we just gave really did what we expected it to, or to find out just what state the system is in at the moment. We don't know why it should be made so difficult, but here are some speculations.

- **We assume that no one wants to know.** Why should they ? People used to put enormous jobs into batch systems and expect them to run without any feedback on what was happening moment by moment.

  (True enough, but batch and interactive disciplines are quite different. Batch is simpler (if only because it isn't plagued by mail messages, and interruptions when the system breaks down, and the temptation to try this quick fix to your problem ....), and the jobs are carefully planned. A good batch system will give you enough information in its job log to follow the state of the system rather precisely, though it might take a little time to work it out.)

- **We assume that the system always does what it's told.** And so we should – it's only a machine, and we expect machines to follow orders without question.

  (But, machine or not, it's a complex system, and it would be good to have at least some acknowledgment that a task has been completed, or that it couldn't be done because of some peculiarity of the system's condition. For example, if we tell the system to remove a file, we'd like to be told if the file couldn't be found, because it probably means we've mistyped the name. An even if we receive this sort of commentary, unless we can get further information when we need it we're left to infer the state of the system from what we think we've done, and it's not easy to remember everything that's happened.)

- **We don't think that we need a response.** Lots of machines don't answer back when they're given instructions – they just get on with the job. Cars don't generally keep telling us what we've done – "Accelerator pressed, wheel turned left, accelerator released, brake applied hard, hit something".

  (But that's misunderstanding the point. Cars already give us quite a bit of information about their internal states – petrol gauge, oil pressure warning, speedometer, etc. – and we can observe quite a lot more by looking out of the window. The traditional computer screen gives us nothing at all. Even a visible cursor probably only means that the terminal is switched on.)

- **We don't know what response to give.** Perhaps this comes a little closer to the point. What we really want to do is to maintain an awareness of the system's state – what's happening now, how many files we have, what's in them, and where they are, whether there is any mail waiting, the state of our account on the system, and as many other things as we can think of. To keep track of everything is difficult, and it certainly doesn't make sense to dump a complete picture of the whole system to the screen every time someone does something.

What can be done to improve matters ? There are some obvious measures, which are implemented in most systems. The most obvious is to provide means to request information on aspects of the system state. That's (almost certainly) universally available for the state of your file system, though even then it isn't always straightforward to retrieve information which goes much beyond the simple lists of files and directories. Just how much of this sort of information is available depends on the operating system's designer (we take the view that operating systems have been designed, though we admit that in many cases there is little enough evidence to support it), but to get it you have to know what to ask for, and how to ask.

To keep track of the state of the system as a whole, other measures are desirable. Here are three possibilities; there are certainly more, but these cover the ground.

SESSION LOG.

A session log is a permanent record of what happened during the computer session, which might be a period at a terminal or a traditional batch job. It will include information about which programmes were executed, and what they did. The level of detail varies from system to system, but a good session log can be very informative. These are very useful in diagnosing things that have gone wrong.

HELP.

One of the functions of a help system might be to make available information on its parent operating system's current state. We shall have more to say on this topic in the chapter *SOURCES OF INFORMATION* .

VISIBLE STATE.

An obstacle faced by designers of early systems, and particularly of the early interactive systems, was the sheer difficulty of moving information about. Everything had to travel along communications paths which were commonly restricted to a few thousand, or even a few hundred, characters per second, and which required considerable effort to administer. The first terminals followed the Teletype conventions, and could only write forwards and down. This established a pattern which was commonly followed in the interests of maintaining compatibility with older equipment, even when character-addressable terminals became available. With such restrictions, any sort of continuous display of the system's state was out of the question. More enterprising systems used character-addressable terminals to good effect, and achieved informative displays, but the enforced character matrix made it hard to implement flexible displays which could give sufficient information.

Modern fast communications and high-definition pixel-addressable screens have completely changed this part of the picture. It is now possible to move large quantities of information around fast enough to be useful (though it's still expensive if you're a long way away), and to use different areas of the screen selectively. Because of these changes, we can now use part of the screen for working and part to display system information, if we want to. It is common to display significant parts of one's file system as a set of windows while one is working on some other task, and the – also fairly recent – ability to move at will between different activities makes it possible to break off from a task to find out information which might not be currently displayed.

COMPARE :

Lane and Mooney[INT3], Chapter 4.