Computer Science 773

Robotics and Real-time Control

PEOPLE AND MACHINES

Most (all ?) control systems operate in two directions; as well as controlling machinery of some sort, they control the machinery *for* some other entity, so they must communicate both with the machinery and with the other entity. As all machinery is built to serve people, directly or indirectly, the other entity is often a person, though in a large system the immediate superior of a controller might be another, supervisory, controller.

Communications between computers are usually well defined, and not particularly difficult to design and construct; a certain number of message types of known form must be specified, and appropriate communications media and interfaces must be provided. Interfaces between computers and people are a very different matter.

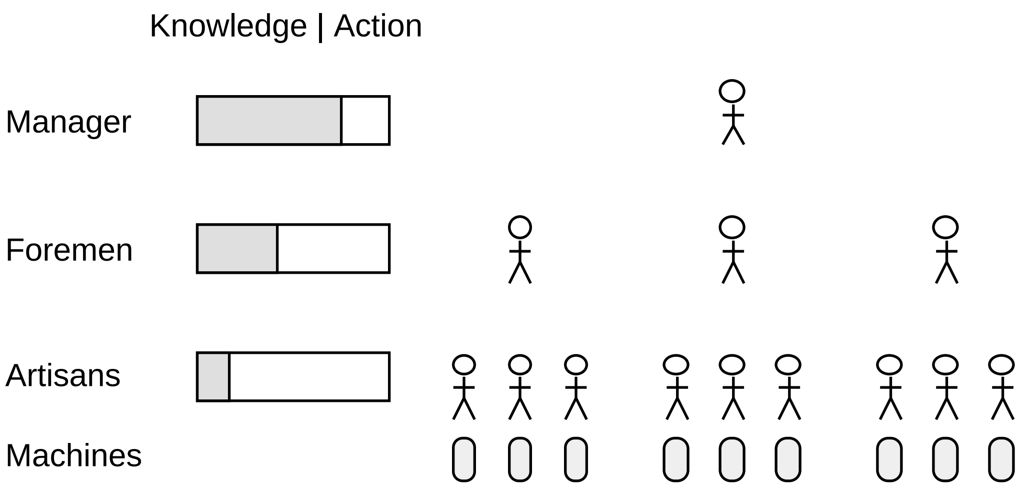
There are at least two reasons for this difficulty. The first is that people are not very good at understanding bit streams, so some effort must be put into presenting the information in ways that we can understand. Printed words and numbers are better than bits; graphical displays – gauges, dials, etc. – might be better still. The second reason is that the people commonly don't really want the raw data which come directly from the plant; they want some rather more abstract information, which depends on the raw data but not necessarily in a simple way. For example, an engineer in charge of a chemical reactor might be able to find out fairly easily the temperature and pressure of the reacting mixture, but is much more interested to know whether the mixture is likely to explode in the next few minutes.

Both those examples are in terms of system output to the controller – the system is presenting information in both cases. Exactly analogous comments apply to the input operations, where the controller must exercise control over the system. For input, we might prefer to turn knobs and push sliders than enter text, and certainly prefer text to entering bits; and we want to be able to give an instruction like "reduce the rate of addition of acid by 20%" rather than "shut off valve 53; run pump motor 22 at half speed".

There are therefore significant problems in designing a good control interface to a machine. On the output side, it is necessary to decide what should be presented, and how to present it; on the input side, what must be controlled and how it should be managed. We rub along fairly well most of the time with rather bad interfaces, because people are really rather clever and able to make the connections between what they have and what they want to do. This gets more difficult as the machines get more complicated, and as responsibility for detail is absorbed into the machine, for in these circumstances the human controller might no longer understand what the machinery is doing well enough to make the necessary connections. Two cases illustrate this point.

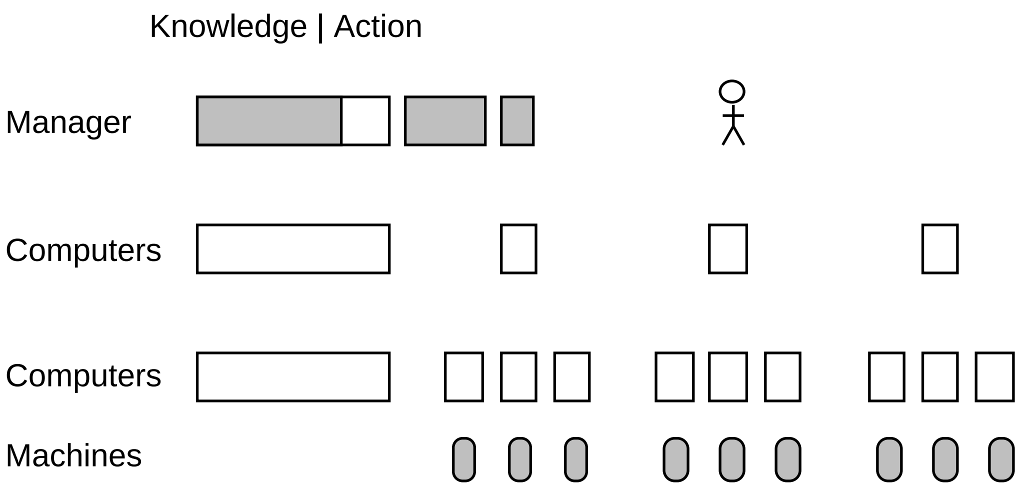
AUTOMATING LARGE SYSTEMS.

The traditional factory is run by a hierarchy of people. This diagram shows a very schematic impression of the structure.



Machines appear only at the bottom level of the hierarchy, and people throughout the system have a lot of knowledge of different kinds about what goes on. (Notice that the diminishing ratio of knowledge to action going down the hierarchy is at least as much a reflection of the greater involvement of the lower levels in the active running of the factory as a suggestion that they have less knowledge.)

The significance of the separation of knowledge from activity becomes clear when we consider the effect of automating the factory. It is very much easier to automate activity than to automate knowledge, so the result can be seen like this :

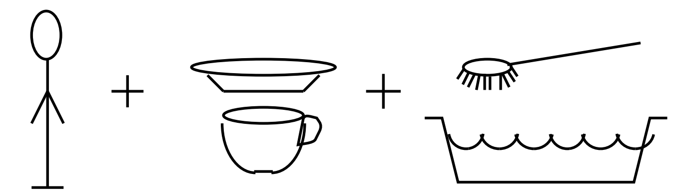
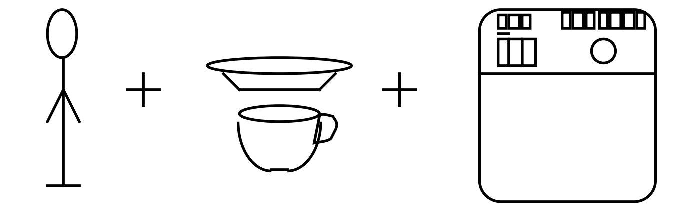


The computers probably run the machinery much more efficiently, but they don't understand it. The artisan will notice if a valve is leaking, or a tool is becoming blunt, or the material is of poor quality, and knows enough to do something about it; similarly, the foreman notices that work is not running smoothly through the department, or that completed work is not being removed quickly enough, and also knows enough to do something about it. When the people are removed, either these functions must also be replaced by machines, which is very hard, or there must be provision to draw such matters to the attention of such people as remain in the system – who must now have a much more detailed understanding of the whole factory than was previously necessary.

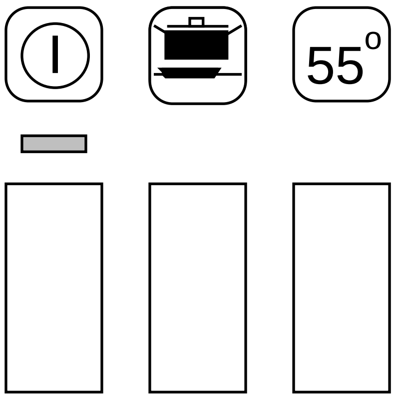
MECHANISING SIMPLE TASKS.

A different, though related, difficulty is found when replacing simple manual tasks using readily comprehensible equipment by much more complicated, and therefore much less comprehensible, machinery. Once again, the root of the problem is that the people who used to use the simple equipment are much more intelligent than the computers which drive the more complicated machines.

Domestic machinery offers many examples of this type. For instance, consider the difference between traditional washing up and a new dishwasher.

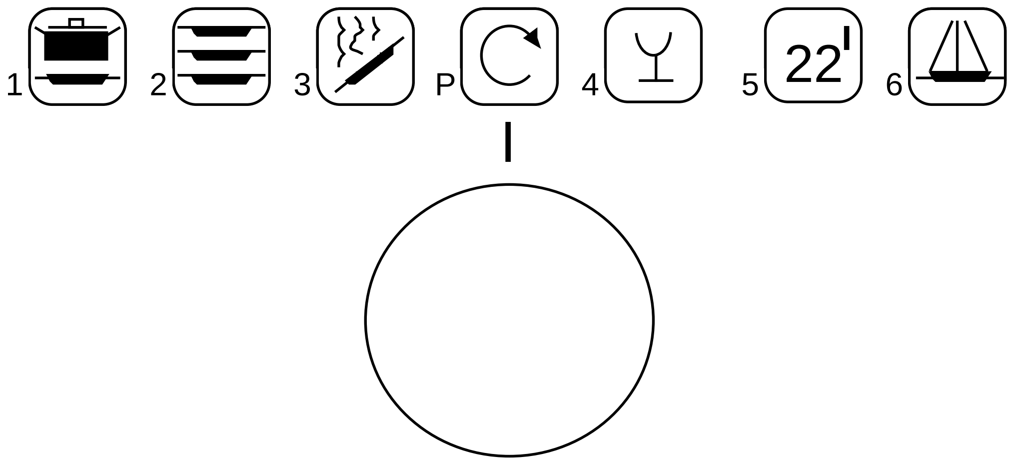
Before		Comprehension -> Action
After		?

Here's a bit more about the dishwasher. It's a real one, and I know it well. The left-hand group of controls sketched on the diagram has to do with the major features of the washing operation. It looks something like this in more detail :



The top row of three squarish things that look like icons are indeed icons, and do nothing else. The rectangles in the bottom row are push-buttons. The leftmost button is the on-off switch, and the small shaded rectangle is a pilot light, which shines whenever the switch is on and there is mains power. You press the rightmost button if you want a coolish wash. What does the middle one do ?

The right-hand group of controls selects what part of the wash cycle you use. It looks more or less like this :



Again, there's a row of icons, above a knob which rotates. What do they mean ? There's a graduated scale on the edge of the knob, with some numbers here and there at uneven intervals. The numbers next to the icons refer to the numbers on the knob. What does the P refer to ?

Finding this essentially incomprehensible, we learnt how to work it from the manual (six European languages, with only one set of diagrams at the end). We set the knob to 2 and press the on-off switch. After a while it stops, and we press the on-off switch again to switch it off, and unload the dishes. We did that once with a not-very-dirty load, and went out after switching it on. On return, it was stopped, and cool, so we emptied the dishes without looking at them very closely. Next time someone got a glass from the cupboard, he found that it wasn't very clean. Neither was the next glass, nor the next. On closer inspection, a lot of other dishes weren't up to our usual standard either.

We eventually found that on switching on the machine, the knob had been set to one gradation before the 2 mark, so no washing at all had happened. The pilot light doesn't switch off automatically at the end of the wash, so there was no indication that anything was amiss. Yes, we should have been more careful – but the design could have been improved.

I don't think that our dishwasher has a computer in it. How could you improve the interface if it had ?

ANSWERS TO SOME OF THE QUESTIONS.

- The instructions for the middle knob on the left-hand panel say "use this button to intensity (sic) the normal wash cycle ... when washing a complete load of very soiled dishes and pots. This option adds hot water to the prewash cycle and an extra rinsing cycle".

- It doesn't tell you what the icons on the right-hand panel mean. By inference from what it does tell you, they mean (left to right) long wash (for dirty pots), shorter wash (for cleaner pots), rinse, ?, light wash (intended for glassware), fast wash and rinse, short cold wash.

- I have no idea what P means.

COMMENTS.

The object of the dishwasher discussion isn't just to laugh at my dishwasher, nor at my family's incompetence. It's intended to illustrate the potential for misunderstanding at the human-machine interface, and two points are notable : first, even quite simple machines can be less than easy to operate; and, second, it isn't really difficult to make their interfaces easier to use. (Getting a perfect design is harder, but some improvement is often quite straightforward.) Having computers in devices makes it easier still, and quite possibly cheaper too.

At the much higher level of complexity in the factory, the problems are not as different as one might expect, because they are still essentially human problems. Again, the manager's difficulties spring from lack of understanding of all the workings of the system, and lack of sources of information about potential errors. (And a lot of other things; using a dishwasher really is easier than managing a factory, but I'm concentrating on the provision of information.)

How can we make interfaces work better, in that they are easier to use and harder to use wrongly ? As we only build the interface machinery, and not the people, the answer must lie in the design of the machinery. It isn't a simple answer, or (one would hope) more people would do it, but some parts of it really are quite easy at the human level. For example, it should be clear, in one way or another, what each control does, and it's exceedingly helpful to have some feedback, in the form of an indication that your control action has had some effect. The principles hold for the manager, too, but they're a lot harder to implement at that level.

To find out how to do better, we have to know a lot more about the two parties to the interface – the people, and the machines. This leads us into some fascinating fields like ergonomics and cognitive science which unfortunately have not much business being in the 773 course, so we're not going much further along this track. If you're interested, I strongly recommend a book by Donald Norman called *The psychology of everyday things* (Basic Books, 1988). It's very readable and highly entertaining, and contains much real food for thought.

Alan Creak,
March, 1998.