Computer Science 773

Robotics and Real-time Control

HIERARCHICAL SYSTEMS

Hierarchical systems are common in computer control and other real-time applications for all but very simple cases. The general pattern is for a high-level process to be associated with one or more lower-level processes in some sort of supervisory capacity; not uncommonly, the lowest-level processes run on different computers (possibly PLCs), typically close to the machines with which they interact, while the high-level process which controls them is elsewhere (typically running on a desktop-scale microcomputer).

The idea of a hieracrchy was there from the start in the DDC systems we saw earlier (see *WHAT IS CONTROL ?*), but the distributed system wasn't; all the processes ran in a single mainframe machine, because that's all there was. This is about the worst possible way to start building control systems, because you have to deal with a very complex system without being able to isolate its components.

There are several ways in which this hierarchical organisation is used, and it is not unusual to find several of these active in a single system. Here are some brief descriptions to illustrate the range of functions covered by such systems. They are in no order other than alphabetic by name.

ADMINISTRATIVE HIERARCHY.

The high-level machine is used for planning and scheduling. It might be linked to the organisation's management system, which sets long-term goals and future development plans, and monitors the levels of demand and stock to determine what to do next. This leads to *Computer-Integrated Management* (CIM).

COMMUNICATIONS HIERARCHY.

The high-level process deals with communications between the lower-level processes, and perhaps between these processes and the world outside. Communication can be within processes running on a single processor, between processors and external devices connected to a common bus, or over a wider area using some form of network communication.

CONTROL HIERARCHY.

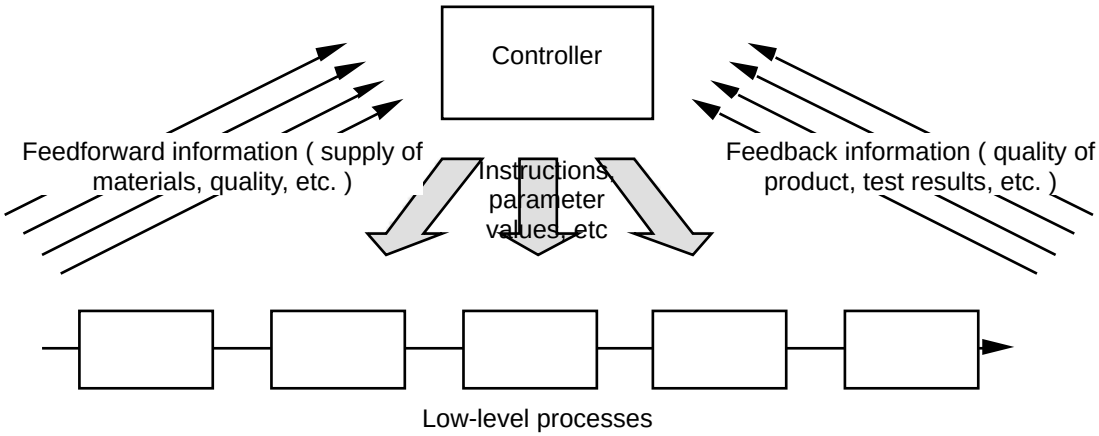
The high-level process exercises overall control over the lower-level processes. This sort of control is almost unavoidable in large control systems, if only because much of the interaction between components usually occurs at a local level and it would be wasteful to handle it all by a central process.

There are several varieties of control hierarchy. A common example is sequence control, where a central sequencer might send control signals to many separate devices, of which several might be driven by programmable logic controllers or by their own embedded computer systems.

A second example is a self-adjusting optimal control system, though these are often not considered as hierarchical. In such a system, a high-level process adjust parameters of

a lower-level process which is in direct control of the machinery in order to improve its performance as conditions change. The high-level process might be driven by observations of changes in external conditions (effectively feed-forward control) or by observations of the machinery's performance, involving some element of supervisory hierarchy and feedback control.

This diagram illustrates some of the relationships between these activities. The feedforward and feedback information could equally be regarded as examples of supervisory control.



RESOURCE HIERARCHY.

The high-level machine acts as a library of software, data, and other resources which might be needed by the lower-level machines from time to time. The library might include diagnostic software in case of failure, alternative programmes for use in manufacturing different products, development aids, and so on.

A common pattern of development which falls into this class is to develop software on a large machine, perhaps using simulators and extensive monitoring and diagnostic tools, then to store the software until required by much less elaborate lower-level machines.

SUPERVISORY HIERARCHY.

The high-level machine monitors the operations of the lower-level machines, constructing reports (typically directed to some still higher level) on work in progress, progress of work (not the same thing !), availability of machines, indications of faults or wear and tear, and generally the state of the plant.

Supervisory activity very commonly goes with control, if only because the information used in the supervisory activity is essentially the same as that required to control the lower-level system. Roughly speaking, in supervision, information flows up the hierarchy, whereas in control instructions flow down.

SOME GENERAL COMMENTS.

Other things being equal, the lower the level, the harder the real-time. The hierarchy serves to soak up urgency, with local dedicated processors provided to deal with very

urgent matters, and less urgent decisions handled by separate processes. Right at the bottom, there might be continuous control of individual plant parameters, but almost all the higher-level control is sequencing.

It might be more precise to say that the higher level control is iterative, as almost all decisions must be reviewed from time to time. In this view, everything is "really" continuous control, and the difference between the low-level continuous control and the plant management reduces (almost) to the cycle time; the speed of a motor might be checked every millisecond, while the range of products manufactured might be checked every year. That's perhaps stretching an analogy rather far, but it does emphasise the essential unity of the enterprise.

Alan Creak,
March, 1998.