Computer Science 773

Robotics and Real-time Control

SYSTEM DESIGN

WHAT HAS TO BE DESIGNED ?

In principle, everything; in practice, it isn't uncommon to be changing an existing design rather than beginning from scratch, so it might be that only part of the system need be designed. If so, the new components are subject to constraints from the requirement that they must fit in with the existing parts. Engineers do a lot of development in this sense; as requirements change, the new demands are often more economically satisfied by alterations to existing products and production facilities than by starting anew.

Just what constitutes "everything" depends on the context. I'll talk mainly about manufacturing systems, because I know more about them, but similar considerations apply to aeroplane control systems, power generation and distribution, chemical plant, railway management, and other sorts of control system.

In manufacturing, three components of the system must be designed : the plant, the process, and the product. These three design problems interact strongly :

- The design of the plant, including the hardware provided and its geography, constrains the operations which can be provided and the sequences of operations which are sensible.
- The design of the product constrains the operations required for its manufacture and their sequence.
- The design of the process constrains the design of both plant and product.

As all three are intimately connected by the purpose of the system, which is to manufacture the product, these interactions are hardly surprising, but it does mean that it isn't sensible to try to design any individual component alone. The only component which it makes any sense to design in isolation is the product, but even here detailed design might take into account plant considerations through group manufacturing techniques or simplified assembly sequences.

The dependences become noticeable when design details are considered. For example, when setting up the plant for the first time, it can be tailored to the specific requirements of the product, but when redesign to accommodate alterations in production method or product is later necessary, the existing plant hardware might strongly constrain the changes which are possible.

DIFFERENCES FROM CONVENTIONAL DESIGN PROBLEMS.

The design of real-time control systems is rather different from that of conventional computer problems. The differences are not so great as to rule out the design methods used for other systems, but they are real enough to warrant at least some changes in emphasis in the application of the methods. There are at least three (related) reasons for the differences.

First, the desired product never appears in the computer programmes. The solution to a mathematical problem, the result of a database search, a payroll, all appear in some form or other in the programmes which are used to produce them, but a washing machine never appears in the control programme which manages its production line. Instead, the programme is only indirectly connected with the product through the manufacturing processes used. In this, there is some resemblance to the programmes for a transaction processing system, but even there the final result – a transfer of funds, or the transmission of a message – has a direct representation within the computer system.

Second, the real function of the control programme is not to make a product but to maintain the state of several machines in such a way that they perform a set of operations which (invisibly to the computer) result in the manufacture of a product. This differs from the usual model of computing, where a programme has to change the state of its system from one including the data to one including the results.

Third, the real-time constraints must be observed, and have no counterpart in most (all ?) traditional design techniques.

Because of these differences, the argument from required product to programme is not as simple as in conventional programming, and design is correspondingly more complicated. Well known methods might not be directly applicable, because they depend on assumptions which are not satisfied; for example, top-down approaches are difficult to transfer directly because there is no simple top. Top-down methods are also particularly inconvenient for timing considerations, for, while they are effective in getting the right code in the right place, timing is very much a bottom-up phenomenon. Methods can often be adapted satisfactorily.

HOW TO DO IT.

The general features of design are much the same as for conventional programmes. The requirements for the system must be stated, then a specification of its expected performance drawn up. This can then be developed into a workable programme structure and eventually encoded. The primary requirements so far as the computing is concerned are determined by the engineering considerations which govern the plant in question; generally, these might include available data and required outputs, timing constraints, and other such basic matters.

There are several approaches to design which I don't intend to present in detail; there are some examples in the textbook. In the sheet *PROGRAMME DESIGN*, I describe a technique which comes somewhere at the end of the specification stage, and results in a detailed system specification and the beginning of system documentation. It's interesting particularly because of the emphasis on formality which is evident throughout. The method is oldish, but the principle remains good.

Alan Creak,
March, 1997.