A PETRI NET APPROACH TO DISCRETE EVENT CONTROL

M J Denham

In this talk, I will describe a theory and procedure for the analysis and synthesis of supervisory controllers for discrete-event systems such as computing, communication and manufacturing and manufacturing systems. The approach described is based on the use of a Petri net model of the discrete-event system.

It is clear that there exists a need for the development of a comprehensive and effective theory for the control of discrete-event systems. A wide variety of systems display discrete-event dynamic behaviour:

* Computing systems - programs, operating systems, human/computer interfaces
* Communication systems - networks, protocols
* Manufacturing systems - batch production, FMS

In addition, many systems display a combination of discrete-event and continuous-time dynamic behaviour, eg systems under some kind of knowledge-based supervisory control.

For supervisory control of discrete-event systems, a method is required for specifying, analysing and verifying the behaviour of what is generally a large set of interacting, concurrently operating subsystems and synthesising a controller to coordinate and direct this behaviour according to specified constraints/performance measures.

In this talk, I shall be concerned with one aspect of this problem, that of controlling the logical behaviour of such systems. This is in cotrast to approaches which aim at controlling the system to meet some quantitative measure of performance, eg throughput. By controlling the system's logical behaviour, I mean controlling the sequence of states and/or events occurring in the system to ensure that the correct sequence takes place under all operational conditions, eg in order to ensure absence of conflicts, freedom from deadlock, satisfaction of safety and/or real-time constraints, etc.

The initial requirement for a model of the logical behaviour of the system leads one naturally to consider those models which have been used to describe discrete-event systems, predominantly comuting and communication systems, in the past. State transition models such as those developed by Hoare [1] and Milner [2] for describing concurrent computing processes have been used successfully as the basis of a control theory for discrete-event systems by Wonham [3]. In this talk, I will describe an approach based on the use of the Petri net model, which is closely related to the state

Department of Computing, Polytechnic South West, Plymouth, UK

transition model, and which has been used extensively to descibe the behaviour of a wide range of discrete-event systems. It is a common model in the area of manufacturing systems, which was one motivation for its use here. It also has the advantage of a clear, concise and readily understandable graphical representation which perhaps underlies its popularity in modelling complex manufacturing systems. Most importantly, it is amenable to both analysis and verification procedures and, as will be seen, to a limited form of synthesis.

The ability to analyse and verify the static logical behaviour of a Petri net model is well documented, eg [4]. By this, I mean the ability to specify a constraint on the instantaneous state of the model and prove that this constraint is satisfied for all legal states in the model's behaviour. This approach can be used to ensure that the model, and by inference the system, is, for example, free from deadlock. In order to synthesise a controller which ensures such deadlock-free behaviour for an arbitary model or collection of interacting models, it is necessary to have a theory about how such a property is affected when two discrete-event models, ie those of the system and the controller, are connected in some form of supervisory control structure. In this talk, I will show that for this category of behavioural constraints, such a synthesis theory can be developed. This is based theoretically on viewing the Petri net model as a two-sorted algebra and the composition of two models in a control structure as a natural product for this algebra. It is shown how the invariant state relation for each sub-model is propagated through the composition operator to define the constraint on the controlled system model.

For dynamic logical constraints on the behaviour of the model, it is necessary to use either language theory, as in [3], or a form of temporal logic, as in [5]. The latter approach, which allows the constraints to be stated explicitly, will be described in the talk. The application, in theory, of this approach to Petri net models is straightforward. However, the approach is computationally intractable for all but simple systems, and a modular approach based on composition operators is required.

Finally, I will show how this approach leads naturally to an object-oriented approach to the specification and design of discrete-event systems and to the software necessary for their control. Recent work [6] has shown how such software can be partially generated automatically from the high-level specifications embodied in the Petri net model of the controlled system, thus leading to a comprehensive and effective CACSD system for discrete-event control.

References

1. Hoare, C A R : "Communicating sequential processes", CACM, vol 21, no 8, pp 666-677, 1978.

2. Milner, R : "A calculus of communicating systems", Lecture Notes in Computer Science 92, Springer-Verlag, 1980.

3. Ramadge, P J & Wonham W M : "Supervisory control of a class of discrete-event processes", SIAM J. Control & Optimisation, vol 25, no 1, pp 206-230, 1987.

4. Peterson, J L : Petri Net Theory and the Modelling of Systems", Prentice-Hall, 1981.

5. Ostroff J S : "Temporal logic and extended state machines in discrete-event control", in Advanced Computing Concepts and Techniques in Control Engineering, M J Denham & A J Laub (eds), Springer-Verlag, 1988, pp 215-236.

6. Bruno G & Marchetto G : "Process-translatable Petri nets for the rapid prototyping of process control systems", IEEE Trans Soft Eng, vol SE-12, no 2, pp 346-357, 1986.