CONTROL OF DISCRETE EVENT PROCESSES

Michael J. Denham

School of Computing, Kingston Polytechnic
Kingston upon Thames, Surrey, U.K.

SUMMARY

This paper considers the problem of devising an
adequate theory in which the behaviour of flexible
manufacturing systems, at the level of coordination and
control of a set of interacting, communicating
concurrent processes, can be modelled and analysed.
Existing approaches to a solution are reviewed and the
outline of a proposed new approach based on some recent
concepts such as object oriented systems is described.

INTRODUCTION

There has been an increased interest in the last few
years in the study of the control of discrete event
processes, perhaps owing to the potential application
in distributed computer networks, communication systems
and advanced computer controlled manufacturing systems.
Approaches taken to the problem by control theorists
include perturbation analysis techniques[1] and algebraic
methods based on the theory of regular languages[2].
There is also a related activity by theoretical
computer scientists to develop theories and methods for
representing and analysing systems which display
discrete event behaviour. Such systems include computer
operating systems, multiple processor systems and
communication protocols. In each case, the objective is
to ensure that the system behaviour conforms to some
prespecified pattern. For a system which is composed of
a number of interacting processes, the problems include
coordination and synchronization of the actions of
individual or sets of processes, controlling the access
to shared resources and ensuring safe operation of the
system under abnormal conditions, e.g. process failure.

In the case of a flexible manufacturing system, the
operational control of the system can be divided into
three levels. At the lowest level, individual processes
are directed by local controllers to carry out a
specified task in an autonomous manner. The problem of
coordinating these individual concurrent actions is
that of the next level of controller, which initiates
such actions and monitors their execution. It also
receives input from the highest level controller, the
scheduler, which determines the overall sequence of
tasks to be carried out by the system. The problem
which we are concerned with is that of representing and
analysing the behaviour of the system at the middle
level of coordination of the constituent processes and
their action in response to a request to perform a
specified schedule. The primary requirement is
therefore for a theory in which the behaviour of a
number of concurrent, interacting discrete event
processes can be represented and analysed and which
provides a basis for the development of techniques for
synthesising the behaviour of such systems.

EXISTING APPROACHES TO THE PROBLEM

The theory of Petri nets[3] has formed an important basis
for the modelling and analysis of such systems for
several years, in particular computer and

communication systems behaviour. More recently, Petri
nets have been used to model flexible manufacturing
systems[4]. Simple state transition networks, and the
associated theory of regular languages and finite state
automata, have also been used to describe concurrent
system behaviour. In particular, a number of results
have been obtained recently on the structural
properties of controlled discrete event processes
modelled in this way[5,6]. The approach uses a control
mechanism in which a controller process monitors
transitions in the controlled process and effects
control by enabling or preventing transitions in the
latter. The results obtained are analageous with those
of geometric control theory, in particular the
existence of and method for synthesising a "minimally
restrictive" supervisory controller, which meets
certain constraints such as the absence of deadlock in
the controlled system.

There are fundamental problems associated with both
these approaches. Firstly, their use in modelling the
behaviour of any reasonably complex system leads to a
cumbersome model which is difficult to understand
conceptually and for which the computational
requirement for exhibiting and analysing system
behaviour increases exponentially with the size of the
system. Secondly, the production of a computer program
which reliably and accurately reproduces the behaviour
of the net theory based model is an ad hoc process
which is prone to human error. In other words, there is
no associated programming language which has a formal
semantics in the same theory, or one directly related
to that in which the model was created. It should be
noted that the computational and complexity problem has
been tackled in the state transition network appraoch
using a modular approach, but this is only effective in
certain restricted cases[6].

Whilst Petri nets have been used extensively for
modelling computing systems, more recently there have
been a number of new developments within the computer
science community of alternative theories and methods
for representing concurrent processes and their
behaviour. Perhaps the most notable amongst these are
those techniques based on temporal logic[7] and Milner's
calculus of communicating systems (CCS)[8]. The
application of temporal logic to discrete event control
problems has recently been considered[9], and the
substantial body of knowledge which has been generated
in recent years by theoretical computer scientists in
this area should prove useful in pursuing this
approach[10]. However there are significant structural
analysis and synthesis problems to be solved, in
particular modular techniqes for system synthesis.
Also no programming language for concurrent systems has
its semantics defined in a temporal logic framework,
creating problems in the implementation of the required
controlled system behaviour.

In the last few years CCS has been applied in
particular to modelling communication protocols, e.g.
for OSI networks, and a language, LOTOS, based on CCS

has been designed for writing system descriptions in this formulation. The use of CCS for modelling other types of communicating systems, including manufacturing systems, is intuitively appealing, since the approach taken is to represent the system as a set of agents or processes which interact or communicate through specified input or output ports. The observable behaviour of each individual process is represented by a behaviour expression, which is essentially derived from the operational behaviour of the process expressed as a finite state automata or state transition network. Specific transitions are linked to input or output ports and processes interact through the interconnection of ports. The theory supplies a set of operations for the composition of the behaviour expressions of the processes, in relation to their concurrent operation and specified interconnection structure, to produce a behaviour expression for the system. This expression can then be analysed to determine various structural properties of the system behaviour, for example that the system does not enter a deadlock situation or that there is a fair access by all processes to shared resources.

At present, it appears that the analysis of systems described in CCS form is restricted to those systems in which the interaction is simply that of synchronization of transitions in individual processes. In such a system, when a process wishes to perform an action which must be synchronized with an action in another process, it signals that process through its corresponding output port and waits until it receives a responding signal which indicates that the related process is in a state to allow the synchronizing action to take place. The two actions then occur in the respective processes. This can be extended to more than two processes. In this simple form of communication, the exchange of data between processes is not permitted. However, a preliminary investigation of the application of the CCS theory to modelling supervisory controllers for discrete event systems, as defined by Wonham and Ramadge[5], indicates that many of the results obtained by them on the structural properties of such systems, can be restated in this framework. It is not yet clear whether the inherently modular form of systems described in CCS will provide any further insight into solving the computational problems associated with the synthesis of controllers previously experienced. It is clear that application of the composition rules of CCS for systems of any real complexity leads to large behaviour expressions for the system.

PROPOSED ALTERNATIVE APPROACH

From the above review of existing approaches, and a knowledge of the requirements of the modelling and control analysis and synthesis problem, it is clear that there are two major criteria for an adequate theory. The first is that it must allow the model of the system to be constructed in a naturally modular way, i.e. the model must be able to be built from objects which are in correspondance with natural objects within the system, for example production processes. A set of rules for composing these objects must be provided, together with a method for reducing the complexity of the composite model, retaining only those aspects which are necessary for describing and analysing the behaviour of the system for the purpose of controller design. The theory must permit the development of both analysis and synthesis methods. Secondly, it must be possible to design a language which can be used by the system designer to create the model, and which has the underlying theory as its formal semantics. This implies that the language is itself object orientated, i.e. that the language can

directly refer to the naturally occuring objects in the system and their interconnection.

At present there are few object oriented languages which can model concurrent system behaviour. One of these is EPSILON, which has been designed as a system description language, with manufacturing systems as one envisaged area of application[11]. A disadvantage of the language is its complex syntactic structure. A strong advantage however is that its semantics can be defined in terms of the theory of coloured Petri nets[12]. These have the same descriptive power as Petri nets but are more concise. There has also been recent work to apply coloured Petri nets in the control of flexible manufacturing systems[13]. Early investigations into this approach indicate that it could provide the theoretical basis for the required control analysis and synthesis methods to be developed and current work is proceeding in this direction[14]. Unfortunately, shortage of space has prevented any presentation here of the theoretical development of this approach.

REFERENCES

1. Ho, Y.C. and Cao, X., "Perturbation analysis and optimization of queueing networks", Journal of Optimization Theory and Applications, 1983.
2. Ramadge, P.J. and Wonham, W.M., "Supervision of discrete event processes", Proc. 21st. IEEE Conf. on Decision and Control, Dec. 1982, pp. 1228-1229.
3. Peterson, J.L., Petri Net Theory and the Modelling of Systems, Prentice-Hall, Englewood Cliffs, 1982.
4. Dubois, D. and Stecke, K.E., "Using Petri nets to represent production processes", Proc. 22nd. IEEE Conf. on Decision and Control, Dec. 1983, pp. 1062-1067.
5. Wonham, W.M. and Ramadge, P.J., "On the supremal controllable sublanguage of a given language", Proc. 23rd. IEEE Conf. on Decision and Control, Dec. 1984, pp.1073-1080.
6. Wonham, W.M. and Ramadge, P.J., "On modular synthesis of supervisory controls for discrete event processes", Proc. Int. IEEE Conf. on Computers, Systems and Signal Processing, Bangalore, Dec. 1984, pp.500-504
7. Manna, Z. and Pnueli, A., "Verification of concurrent programs: the temporal framework", in Boyer and Moore (eds.), The Correctness Problem in Computer Science, Academic Press, New York, 1981, pp.215-273.
8. Milner, R., A Calculus of Communicating Systems, Lecture Notes in Computer Sci. 92, Springer-Verlag, 1980.
9. Thistle, J.G. and Wonham, W.M., "Control problems in a temporal logic framework", Systems Control Group Report 8510, Dept. of Electrical Eng., Univ. of Toronto, Aug. 1985.
10. Manna, Z. and Wolper, P., "Synthesis of communicating processes from temporal logic specifications", Proc. Workshop on Logics of Programs, Lecture Notes in Computer Science, Springer-Verlag, 1981.
11. Jensen, K. and Kyng, M., "EPSILON. A system description language", Report PB-150, Computer Science Dept., Aarhus University, Oct. 1982.
12. Jensen, K., "Coloured Petri nets and the invariant-method", Theoretical Computer Science, vol.14, 1981, pp. 317-336.
13. Martinez, J. and Silva, M., "A language for the description of concurrent systems modelled by coloured Petri nets: application to the control of flexible manufacturing systems", Proc. IEEE Workshop on Languages for Automation, 1984.
14. Denham, M.J., "A theoretical framework for the control of discrete event processes", Report 8610, School of Computing, Kingston Polytechnic, 1986.