Distributive Control Design for Discrete Event Systems

M.A. de Ridder and M.P. Spathopoulos

Division of Dynamics and Control
University of Strathclyde
Glasgow, UK
E-mail: m_ridder@uk.ac.edrc and m.spathopoulos@uk.ac.strath.vaxa

Extended Abstract

In this paper, we consider the problem of modelling and supervisory control of discrete event systems using finitely recursive processes. The algebra of finitely recursive processes is derived by Inan and Varaiya [2] and [3]. The model of finitely recursive processes is a set of recursive equations

$$X = F(X)$$
$$Y = G(X)$$

where $X = (X_1, \ldots, X_n)$, each component f_i of F has the form

$$f_i(X) = (a_{i1} \rightarrow f_{i1}(X) | \cdots | a_{ik_i} \rightarrow f_{ik_i}(X))_{\mu_i}$$

and f_{ij} and G are constructed using some fundamental operators as building blocks.

Using algebras of processes, we model a discrete event system using an interactive graphical representation of both structure and behaviour based on system decomposition, subsystem interaction and subsystem behaviour. The behaviour of a subsystem of a discrete event system is described in the form of an input/output system, mapping the inputs (processes) to outputs (processes), see De Ridder and Spathopoulos [1] and [5].

The structure of discrete event systems is defined by its hierarchical structure and the connections between the inputs and the outputs of its subsystems. The structure of the discrete event system is expressed in the form of a combination of a decomposition tree and connection diagrams. The modularity embedded in the calculus of finitely recursive processes, enabling reuse of components, makes the interaction between system structure and behaviour possible.

Next, we consider the problem to develop control strategies in order to ensure desirable behaviour of a model. Supervisory control is control in which the occurrence of events can be disabled (blocked) or enabled. However, the design of supervisory controllers for complex discrete event systems can be extensive and elaborate.

Our aim is to employ some ideas of supervisory control distributively based on the proposed discrete event system structure and (sub)system behaviour. The idea behind supervisory control is to design a supervisor which enables or disables events during the evolution of the process so that undesired traces are prevented to occur, see Ramadge and Wonham [4].

However, in the developed theory of supervisory control the synthesis of a supervisor for a discrete event system requires an automaton model of the desired behaviour which can be very difficult to be obtained. Moreover, the desired behaviour obtained as an automaton model may not change dynamically, i.e. may not be time-varying.

In this paper, we consider a supervisory control design when the desired behaviour is given in terms of some logical control objectives. We introduce a logical control objective as the tripple

$$[\beta, A_t, A_f]$$

which means that each event in A_t or A_f should be blocked whenever the boolean expression β evaluates to true or false respectively.

The boolean function β has the form

$$\beta : trY \times M \rightarrow \{true, false\}$$

and depends on the process behaviour (trace s of the process Y) and on the marking function of the process (which is used to describe termination and/or data variables).

Instead of attempting to construct the complete control policy for the discrete event system, we propose a distributive control system where each component of the system is controlled by a local controller following global specifications. Such a local controller can perform dynamic control in the sense that it disables and enables events depending specifically on the process behaviour.

Given a set of logical control objectives, we propose a technique to derive the set of controllable logical objectives corresponding to the supremal controllable sublanguage.

In this paper, we present a technique to derive the distributive control system by constructing the control components and their connections with the system from the set of controllable control objectives. The control components are built using the local and global change operators which are embedded in the finitely recursive process formalism and allow the designer to enable and disable events dynamically using the event function.

The design methods are illustrated using an example of a subway system, which is a case where the automaton based methods fail due to the complexity of the system and the dynamical nature of the legal behaviour.

References

[1] M.A. de Ridder and M.P. Spathopoulos. Interactive graphical representations of structure and behaviour for discrete event systems. In *IEEE International Symposium on*

Intelligent Control, Glasgow, UK, August 11-13 1992.

[2] K. Inan and P.P. Varaiya. Finitely recursive process models for discrete event systems. *IEEE Transactions on Automatic Control*, 33(7):626–639, July 88.

[3] K. Inan and P.P. Varaiya. Algebras of discrete event models. *IEEE Proceedings*, 77(1):24–38, January 89.

[4] P.J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, January 1989.

[5] M.P. Spathopoulos and M.A. de Ridder. Some issues on the modelling and control of discrete event systems using an algebra of processes. In *IMA International Conference on Control,: Modelling, Computation and Information*, Manchester Conference Centre, September 2-4 1992.