Centralized and Decentralized Control of Partially-Observed Discrete-Event Systems: The State of the Art and Some New Results

ARO Workshop 2000 on Intelligent Systems
Sponsored by Army Research Office
8-9 December 2000
The Australian National University
Canberra, Australia

Stéphane Lafortune
Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, Michigan 48109-2122
USA
tel: +1-734-763-0591
fax: +1-734-763-8041
email: stephane@eecs.umich.edu
URL: `www.eecs.umich.edu/~stephane`
September 19, 2000

Extended Abstract

Our objective in this talk is to present an overview of key results on the control of partially-observed discrete-event systems (DES hereafter). Both centralized and decentralized control architectures will be considered. The control framework adopted is that of the theory of supervisory control of DES, initiated by Ramadge & Wonham in the 1980's [4]. We consider a DES modeled by an automaton (or state machine) denoted by G; let the set of event labels in G be denoted by E. Equivalently, the system is modeled by the languages generated and marked by G, denoted by $\mathcal{L}(G)$ and $\mathcal{L}_m(G)$, respectively. The prefix-closed language $\mathcal{L}(G)$ models all the traces of events that the system can execute while the marked language $\mathcal{L}_m(G)$ models those traces in $\mathcal{L}(G)$ that represent, by modeling choice, the completion of some operation or task. The notion of marked language, or equivalently the notion of marked states in G, allows the consideration of blocking (deadlock and livelock) in the analysis of DES.

The automaton G models the uncontrolled behavior of the system. This behavior must be restricted by control in order to ensure that only legal traces of events are generated and that blocking does not occur (or its effect is mitigated if blocking cannot be completely eliminated). Control is exerted by means of a supervisor, denoted by S, that observes the events generated by G and controls the events that G is allowed to execute. The controlled system is denoted by S/G. In order to account for actuation and sensing limitations, the set of events E is partitioned in two ways. Regarding actuation limitations, E is partitioned into $E = E_c \cup E_{uc}$, where E_{uc} is

the set of uncontrollable events and E_c is the set of controllable events. The controllable events are those events that can be enabled or disabled by the supervisor. Regarding sensing limitations, E is partitioned into $E = E_o \cup E_{uo}$, where E_{uo} is the set of unobservable events and E_o is the set of observable events. The observable events are those events that can be observed, or "seen," the supervisor. When $E_{uo} \neq \emptyset$, the supervisor is often denoted by S_P, where the subscript P refers to partial observations.

The control architecture described above is depicted in Fig. 1. The theory of supervisory control of DES has been able to answer many fundamental questions regarding necessary and sufficient conditions for the existence of supervisors that achieve a given legal behavior that captures all the requirements (or specifications) imposed on G. This theory has also led to the development of algorithmic procedures that synthesize supervisors that are guaranteed to be *safe* (in the sense that they result in a controlled behavior that never exceeds the legal behavior) and *nonblocking* (in the sense that there is no deadlock or livelock). The four "key" properties of this discrete-event system theory are: *controllability*, *nonconflicting*, *observability*, and *co-observability* [1]. These properties are stated as language properties, hence independent of any particular DES modeling formalism (e.g., automata, Petri nets, process algebras). The algorithmic procedures that have been developed for testing these properties and synthesizing supervisors are restricted at present to finite-state systems and employ automaton models for the DES (namely, G is a finite-state automaton) and for the legal behavior. In this talk, we will focus mostly on the properties of observability and co-observability.

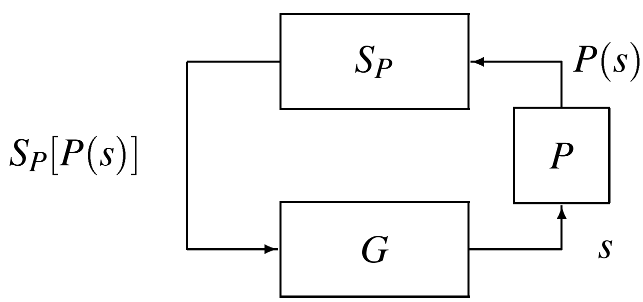

Figure 1: The feedback loop of supervisory control in the case of partial observation.
The projection $P : E^* \rightarrow E_o^*$ hides the unobservable events executed by G from supervisor S_P.

Observability is a key ingredient in the necessary and sufficient conditions for the existence of a supervisor that achieves exactly the legal behavior in the context of Fig. 1. An intuitive explanation of observability will be presented. Observability is easily verified by building the observer of G with respect to the set of observable events E_o [1]. However, the state space of the observer of G may be exponential (in the worst case) in the state space of G. It turns out that there exists a polynomial test for observability [6]. It was also shown in [6] that when the legal language is not observable, the problem of synthesizing a nonblocking and safe supervisor is PSPACE-hard; however, the decidability of this problem remained an open problem. We have recently shown that this problem is decidable [7]. On the other hand, if the desired language is observable with a given set of observable events E_o, one may be interested in finding a subset of E_o of minimum cardinality such that the legal language remains observable. This corresponds to removing sensors that are "redundant" from a control viewpoint. We have recently shown that this problem is NP-complete [10]. However, if more structure is included into the problem, for instance in the context of a probabilistic formulation, then there exist polynomial-time algorithms to minimize the cardinality of the set of sensors (i.e., observable events) [2].

We will then turn our attention to the decentralized control architecture depicted in Fig. 2, where a set of supervisors (only two are shown in Fig. 2), each observing a different subset of E_o and controlling a different subset of E_c, jointly control G. Co-observability is a key ingredient in the necessary and sufficient conditions for the existence of a set of supervisors that together achieve exactly the legal behavior in the context of Fig. 2. An intuitive explanation of co-observability will be presented. Co-observability can be tested in polynomial time [5]. Recently, it has been shown that if the legal language is not co-observable, the problem of synthesizing a supervisor that is both safe and nonblocking is undecidable [3]. This result is somewhat surprising since both G and the automaton description of the legal language have finite state spaces. The problem becomes decidable if the nonblocking condition is relaxed. Consequently, there appears to be some fundamental difficulties that arise when dealing with the nonblocking condition in decentralized control architectures.

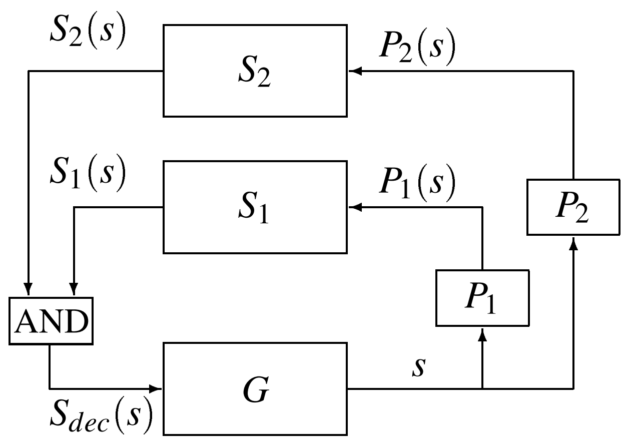

Figure 2: Decentralized control architecture.
This architecture is said to be "conjunctive" as the control actions (enabled events) of the local supervisors are fused by intersection.

We will conclude this talk by presenting a novel decentralized control architecture where the control actions of the individual supervisors are combined in a more flexible manner than in Fig. 2. Namely, the supervisors agree *a priori* on choosing "fusion by union" (of enabled events) for certain controllable events and "fusion by intersection" for the other controllable events, as shown in Fig. 3. This control architecture is more powerful than the purely conjunctive one in Fig. 2 in the sense that a relaxed version of co-observability appears in the necessary and sufficient conditions for the existence of a set of supervisors that achieve a given legal language. This relaxed version is also verifiable in polynomial time [8]. Moreover, the "optimal" (in a sense that can be made precise) partition of controllable events between fusion by union and fusion by intersection in order to guarantee the safety of the controlled behavior can also be determined in polynomial time [9].

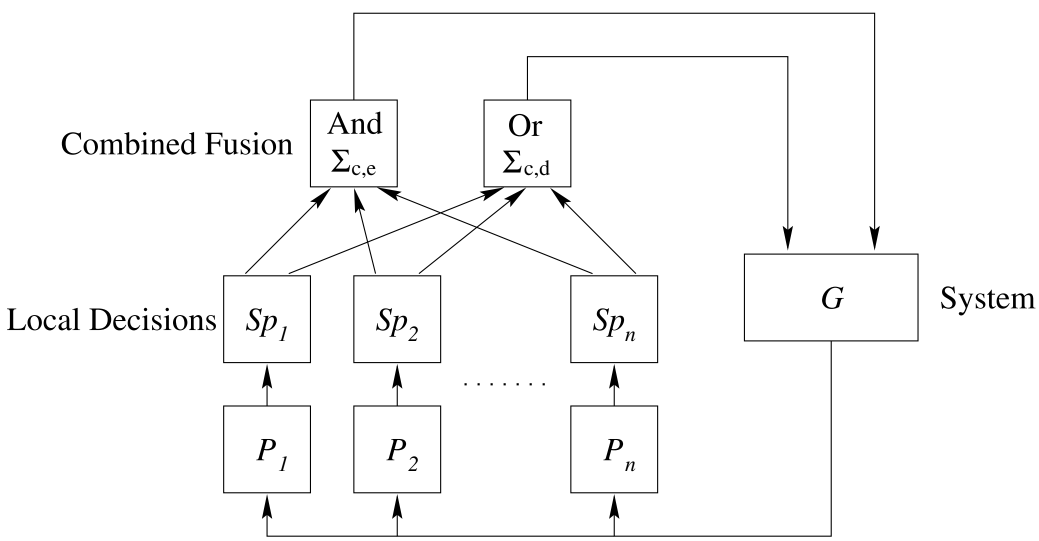

Figure 3: A more general decentralized control architecture.
The local control actions of the individual supervisors are combined by conjunction (intersection of enabled events) for some of the controllable events and by disjunction (union of enabled events) for the remaining controllable events.

References

[1] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.

[2] R. Debouk, S. Lafortune, and D. Teneketzis. On an optimization problem in sensor selection for failure diagnosis. In *Proc. 38th IEEE Conf. on Decision and Control*, December 1999.

[3] H. Lamouchi and J. G. Thistle. Effective control systhesis for des under partial observations. In *Proc. 39th IEEE Conf. on Decision and Control*, December 2000.

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

[5] K. Rudie and J. C. Willems. The computational complexity of decentralized discrete-event control problems. *IEEE Trans. Automatic Control*, 40(7):1313–1318, July 1995.

[6] J. N. Tsitsiklis. On the control of discrete-event dynamical systems. *Math. Control Signals Systems*, 2:95–107, 1989.

[7] T.-S. Yoo and S. Lafortune. Decidability of supervisory control problems with partial observations. Manuscript in preparation.

[8] T.-S. Yoo and S. Lafortune. A general architecture for decentralized supervisory control of discrete-event systems. In R. Boel and G. Stremersch, editors, *Discrete event systems: Analysis and Control*, pages 111–118. Kluwer Academic Publishers, 2000.

[9] T.-S. Yoo and S. Lafortune. New results on decentralized supervisory control of discrete event systems. In *Proc. 39th IEEE Conf. on Decision and Control*, December 2000.

[10] T.-S. Yoo and S. Lafortune. On the computational complexity of some problems arising in partially-observed discrete-event systems. In *Proc. 2001 American Control Conf.*, June 2001. Submitted.