Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Formal Modeling and Synthesis of Event-Transferring Communication among Decentralized Supervisors for Discrete-Event Systems

A. Mannani and P. Gohari

Abstract— **This work proposes to model and synthesize Event-transferring communication among decentralized supervisors for a Discrete-Event System (DES) within the framework of Distributed Supervised Discrete-Event Systems (DSDESs), which was introduced by the authors. This relies on the Polynomial Dynamical System (PDS) representation of DSDESs, which reveals the informational dependencies among supervisors. To serve these dependencies, communication between every two supervisors is modeled by a communication event, whose semantics is defined by a map from observable events of the issuer of the communication to its set of event-encoding variables. Thereby, the synthesis of communicating decentralized supervisors is reduced to the design of these maps using standard algebraic tools. The approach is illustrated through formal synthesis of an information policy.**

I. INTRODUCTION

Supervisory Control Theory (SCT) seeks to minimally restrict the behavior of a DES, called plant, within the language of a given specification by designing a (centralized) supervisor, which disables some of the plant's transitions [1]. In the absence of global observation of the plant's behavior, a finite set of decentralized supervisors, each observing the plant's behavior locally, have to be designed such that their synchronous supervision leads to the same closed-loop behavior which is enforced by the centralized supervisor. If the specification is not coobservable [2], i.e. some illegal plant's moves cannot be distinguished from legal moves by any of the local supervisors which can disable them, the local supervisors need to communicate amongst them to disambiguate their observations and meet the specification.

Inspired by control over networks [3], the study of communication should specify the sender, receiver, content, time, and the order of communication [4]. Most pioneering researches, listed in [5], rely on behavioral formulations and are limited to high-level algebraic structures of DESs. In the authors' viewpoint, this study would be more fruitful if the algebraic structures of the supervisors are utilized in more details. The idea is to encode the states of the corresponding centralized supervisor in a distributed way and represent its observation and control tasks as dynamic and algebraic equations in *symbolic* form. This makes the dynamics-related information amenable to algebraic characterizations and the study systematic, simplified and practically appealing [6].

Symbolic systems and PDSs have been of interest since the early days of SCT to formulate DES *centralized* control problems [7], [8] or to improve the associated computational efficiency [9], [10]. However, we use PDSs to model, synthesize, and analyze communicating *decentralized* supervisors.

In [6], the authors introduced distributed EFSM framework which, using an Agent-wise Labeling Map (ALM), represents the state structure of a centralized supervisor in a distributed way, and captures its observation and control information as *guards* and *actions* over Boolean variables, respectively. DSDES framework, then proposed in [5], employs *guard* and *updating functions*, defined on integer labels. Utilizing the meaningful and compact representation of a state, DSDES framework improves mathematical proofs and computations and, on top of its qualitative-like vantage point to system representation, it can be readily put into concrete implementation using Boolean variables.

This paper continues the authors' work on modeling and synthesis of communicating supervisors within DSDES framework [5]. The content of communication can be (encodings of) states, events, or both. Whereas the first type is discussed in [11], the second type, called *event-transferring* communication, is modeled and synthesized here. Accordingly, communication between supervisors serves to inform the receiver of the event observed by the sender, and is used by the receiver to reevaluate its guard or updating functions. This is modeled as a "communication event," whose semantics is determined as a map from the sender's observable events to its event-encoding Boolean variables. Thus, the synthesis of decentralized supervisors is reduced to the design of these maps. This is illustrated by synthesizing an *information policy*, which is justified on an example.

Section II reviews the DSDES framework of [5] and its PDS representation of [11]. Section III models event-transferring communication and Section IV presents the synthesis of such communication and justifies it.

II. DSDES FRAMEWORK AND ITS PDS REPRESENTATION

Let Σ be a finite alphabet and $L \subseteq \Sigma^*$ be plant's behavior, or simply plant. Consider a network consisting of distributed sensors and actuators as means to observe and control, respectively, the plant's behavior for n supervisors. Denote by $\mathbf{S}_i$ the i'th supervisor in the network, where $i \in I = \{1, 2, \ldots, n\}$. Associate with $\mathbf{S}_i$ observable and controllable event subsets $\Sigma_{o,i}$ and $\Sigma_{c,i}$, respectively, where $\Sigma_{o,i}, \Sigma_{c,i} \subseteq \Sigma$. Thus, from the viewpoint of the i'th supervisor we have $\Sigma_{uo,i} = \Sigma \setminus \Sigma_{o,i}$ and $\Sigma_{uc,i} = \Sigma \setminus \Sigma_{c,i}$. Define $\Sigma_i = \Sigma_{c,i} \cup \Sigma_{o,i}$. Associated with each event σ denote by $I_o(\sigma)$ the set of all sensors which can observe σ, i.e. $I_o(\sigma) = \{i \in I \mid \sigma \in \Sigma_{o,i}\}$. We define

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

The first author is with the ECE Dept., Concordia University, Montreal, QC, Canada, H3G 1M8, amin_man@ece.concordia.ca. The second author was with the same institute, but unfortunately passed away in November 2008.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

the centralized supervisor, denoted by $\mathbf{S}$, to be one which has access to all sensors' observations and can exercise control over all controllable events. For this supervisor we define $\Sigma_c = \bigcup_{i\in I}\Sigma_{c,i}$, $\Sigma_o = \bigcup_{i\in I}\Sigma_{o,i}$, $\Sigma_{uo} = \Sigma \setminus \Sigma_o$, $\Sigma_{uc} = \Sigma \setminus \Sigma_c$, and $P : \Sigma^* \to {\Sigma_o}^*$. $\mathbf{S}$ is modeled by an automaton $\mathbf{S} = (R, \Sigma, \xi, r_0, R_m)$, where R is the finite set of states, r_0 is the initial state, R_m is the set of marked states, and $\xi : R \times \Sigma \to R$ is the partial transition function[1]. Let $\mathbb{N} = \{0, 1, 2\cdots\}$. Denote by $\underline{v} = (v_1, \dots, v_n) \in \mathbb{N}^n$ a vector of n natural numbers and let $\underline{0}$ denote a vector of n zeros. Consider a map $\pi_i : \mathbb{N}^n \to \mathbb{N}$ such that $\pi_i(\underline{v}) = v_i$ which picks the ith component of v, and extend π_i to a map $pwr(\mathbb{N}^n) \to pwr(\mathbb{N})$. The prefix closure of a language $L \in \Sigma^*$ is shown by $\overline{L}$.

A Distributed SDES (DSDES) is denoted by $\mathcal{D} = \{\mathcal{D}_i\}_{i\in I}$, where each quadruple $\mathcal{D}_i = (\Sigma, L, \mathcal{A}_i, \mathcal{G}_i)$ is defined as follows. Σ is a finite set of events (alphabet), L is a (regular) language defined over Σ, i.e. $L \subseteq \Sigma^*$, $\mathcal{A}_i : \Sigma_i \times \mathbb{N}^n \to \mathbb{N}$ is an *updating* function, and $\mathcal{G}_i : \Sigma_i \to pwr(\mathbb{N}^n)$ is a *guard* function. For convenience we extend the domain of $\mathcal{A}_i$ and $\mathcal{G}_i$ to the alphabet of all events. Define $\hat{\mathcal{A}}_i : \Sigma \times \mathbb{N}^n \to \mathbb{N}$ and $\hat{\mathcal{G}}_i : \Sigma \to pwr(\mathbb{N}^n)$ according to: for $\sigma \in \Sigma$ and $\underline{v} \in \mathbb{N}^n$,

$$\hat{\mathcal{A}}_i(\sigma,\underline{v})=\begin{cases}\mathcal{A}_i(\sigma,\underline{v}) & ;\sigma\in\Sigma_i\\ \pi_i(\underline{v}) & ;\sigma\notin\Sigma_i\end{cases},\hat{\mathcal{G}}_i(\sigma)=\begin{cases}\mathcal{G}_i(\sigma) & ;\sigma\in\Sigma_i\\ \mathbb{N}^n & ;\sigma\notin\Sigma_i\end{cases} \quad (1)$$

In the natural recursive way, $\mathcal{A}_i$ is extended to $\hat{\mathcal{A}}_i : \Sigma^* \times \mathbb{N}^n \to \mathbb{N}$. We shall use $\mathcal{A}_i$ and $\mathcal{G}_i$ to denote $\hat{\mathcal{A}}_i$ and $\hat{\mathcal{G}}_i$, respectively. Define a map $\mathcal{A} : \Sigma^* \times \mathbb{N}^n \to \mathbb{N}^n$ recursively as follows: for all $\underline{v} \in \mathbb{N}^n$, $s \in \Sigma^*$, and $\sigma \in \Sigma$

$$\mathcal{A}(\epsilon, \underline{v}) = \underline{v};\ \mathcal{A}(s\sigma, \underline{v}) = \Big(\mathcal{A}_i(\sigma, \mathcal{A}(s, \underline{v}))\Big)_{i\in I}. \quad (2)$$

Associated with each index $i \in I$, a DSDES is equipped with guard and updating functions to capture control and observation, respectively. Control for each $\mathcal{D}_i$ is based upon n-vectors of natural numbers; component i of a vector is updated with $\mathcal{A}_i$.

The semantics of $\mathcal{D}$ is as follows: to each string $s \in \Sigma^*$ a label $\mathcal{A}(s, 0)$ is attached. Thus, starting recursively from ϵ, if s is in the behavior of $\mathcal{D}$ and $\sigma \in \Sigma$ is eligible in $\overline{L}$ after s (i.e. $s\sigma \in \overline{L}$), then σ is "enabled" if the label of s is in the image of σ under the guard function, i.e. $\mathcal{A}(s, 0) \in \mathcal{G}(\sigma)$. When σ is taken, the label of $s\sigma$ is computed according to $\mathcal{A}(s\sigma, 0) = \mathcal{A}(\sigma, \mathcal{A}(s, 0))$. The behavior of $\mathcal{D}$ is a subset of L. A DSDES is obtained by *guarding* events, i.e. limiting their occurrence, based on the *observation* of event sequences of the guarded language. Thereby, a DSDES is equipped with means to *control* and *observe* a given behavior L; in other words, a DSDES may be used to *implement* the control decisions of an already designed supervisor for L, and thus it is suitable to model a closed-loop DES.

Problem 1 *DSDES Control problem*: Let a proper, feasible and admissible centralized supervisor $\mathbf{S} = (R, \Sigma, \xi, r_0, R_m)$ enforce a specification E for a plant $\mathbf{G}=(Q, \Sigma, \delta, q_0, Q_m)$. Design guard and updating functions for each $\mathcal{D}_i = (\Sigma, L_m(G), \mathcal{A}_i, \mathcal{G}_i)$ s.t. $L(\mathcal{D}) = \overline{E}$ and $L_m(\mathcal{D}) = E$. □

[1]Same hold for any recognizers such as $\mathbf{G} = (Q, \Sigma, \delta, q_0, Q_m)$, too

The solution to this problem relies on the assignment of integer vector labels to the states of $\mathbf{S}$ using ALMs. An ALM for $\mathbf{S}$ is a map $\ell : R \to pwr(\mathbb{N}^n)$ such that $\underline{0} \in \ell(r_0)$,

- $\forall r, r' \in R.\ r \neq r' \Rightarrow \ell(r) \cap \ell(r') = \emptyset$, and
- $\forall r, r' \in R, r \neq r',\ \forall\sigma \in \Sigma_o,\ \forall\underline{v} \in \mathbb{N}^n.\ \ \underline{v} \in \ell(r) \wedge r' = \xi(r, \sigma) \implies \exists!\underline{v}' \in \mathbb{N}^n.\ \underline{v}' \in \ell(r') \wedge$
  $[\forall i \in I_o(\sigma).\ v_i \neq v'_i] \ \wedge\ [\forall j \in I \setminus I_o(\sigma).\ v_j = v'_j]$.

The fact that a finite ALM, i.e. one with finite image, exists for every $\mathbf{S}$ [6], paves the way for defining the updating functions associated with $\mathcal{D}$. To this end, a map $\mu : \Sigma\times\mathbb{N}^n \to \mathbb{N}^n$ can be defined such that

$$\forall r, r' \in R,\ \forall\sigma \in \Sigma,\ \forall\underline{v} \in \mathbb{N}^n. \quad \underline{v} \in \ell(r) \wedge r' = \xi(r, \sigma)$$
$$\implies [\mu(\sigma, \underline{v}) \in \ell(r') \ \wedge\ \big(\forall i \in I_o(\sigma).\ \pi_i(\mu(\sigma, \underline{v})) \neq \pi_i(\underline{v})\big)$$
$$\wedge\ \big(\forall j \in I \setminus I_o(\sigma).\ \pi_j(\mu(\sigma, \underline{v})) = \pi_j(\underline{v})\big)].$$

The updating functions can be defined using map μ:

$$\forall r, r' \in R, \forall\sigma \in \Sigma, \forall\underline{v} \in \mathbb{N}^n.\ r' = \xi(r, \sigma) \ \wedge\ \underline{v} \in \ell(r)$$
$$\implies \mathcal{A}_i(\sigma, \underline{v}) = \pi_i(\mu(\sigma, \underline{v})). \quad (3)$$

Proposition 1 *A solution to Problem 1:* For all $i \in I$, let $\mathcal{A}_i$ be as in (3), and $\mathcal{G}_i$ be as follows: For every $\sigma \in \Sigma$

$$\mathcal{G}_i(\sigma)=\begin{cases}\{\ell(r) \mid r \in R \wedge \xi(r, \sigma)!\}; & \text{if } \sigma \in \Sigma_{c,i},\\ \cup_{r\in R}\ell(r); & \text{if } \sigma \in \Sigma_{uc,i}.\end{cases} \quad (4)$$

Then $L(\mathcal{D}) = \overline{E}$ and $L_m(\mathcal{D}) = E$. ■

Computation of the characteristic equation and transition function associated with, respectively, guard and updating functions leads to a PDS representation of the corresponding DSDES. To this end, we use the method of *interpolation polynomials in the Lagrange form* [12]. Let $p \geq 2$ be a natural number, $\mathbb{F}_p$ be a finite field of p integers with addition defined modulo p, x_i ($i \in I$) be a variable taking values from $\mathbb{F}_p$, $\mathbf{x} = (x_1, \cdots, x_n)'$, and $\mathbb{F}_p[\mathbf{x}]$ be the ring of polynomials in the variables $x_1, \cdots, x_n$ and coefficients taken from $\mathbb{F}_p$ [13]. For a function or formula f, by writing $f(\mathbf{x})$ we mean that f can in general depend on some or all of the elements of $\mathbf{x}$. The set of variables on which f precisely depends is denoted by either $\arg(f)$ or explicit listing of such variables. Let x_i be $\mathbf{S}_i$'s private variable, with respect to which all x_j's, $j \in I\setminus\{i\}$, are referred to as *external* variables. For an event $\sigma \in \Sigma$, the polynomials corresponding with $\mathcal{A}_i(\sigma, .)$ and $\mathcal{G}_i(\sigma)$ are denoted by $\mathfrak{a}_i^\sigma(\mathbf{x})$ and $\mathfrak{g}_i^\sigma(\mathbf{x})$, respectively.

Given the graph of a function $f : \mathbb{F}_p^n \to \mathbb{F}_p$ as a set $\mathcal{U} = \{(\mathbf{u}, y) \in \mathbb{F}_p^n\times\mathbb{F}_p \mid y = f(\mathbf{u})\}$, the method of "interpolation polynomials in the Lagrange form" computes a polynomial $q \in \mathbb{F}_p[\mathbf{x}]$ such that $q(\mathbf{u}) = f(\mathbf{u})$. Algorithm 1 in [11] provides a procedure for computing polynomial functions associated with guard and updating functions.

Definition 1 Associated with Proposition 1, for each $\mathbf{S}_i$ and each $\sigma \in \Sigma$ let polynomial equations $x_i := \mathfrak{a}_i^\sigma(\mathbf{x})$ and $\mathfrak{g}_i^\sigma(\mathbf{x}) = 1$ over $\mathbb{F}_p^n$ replace the updating function $\mathcal{A}_i(\sigma, .)$ in (3) and guard function $\mathcal{G}_i(\sigma)$ in (4), respectively. The polynomials are said to *represent* the DSDES if they result in $L(\mathcal{D}) = \overline{E}$ and $L_m(\mathcal{D}) = E$. □

Proposition 2 Let $\mathbf{G}$, E, $\mathbf{S}$, ℓ, be as in Proposition 1. The polynomial equations, obtained by computation of $\mathfrak{a}_i^\sigma$ and $\mathfrak{g}_i^\sigma$ using Algorithm 1 in [11], represent the DSDES. ■

Using this "symbolic" formulation, every DSDES can be represented as a *PDS in state explicit form* [7], where equations associated with updating and guard functions represent the dynamics of the DES and its algebraic constraints, respectively. An updating function, associated with a non-observable event, and a guard function, associated with an uncontrollable event, are identity and unity functions, respectively, whose removal yields the following PDS.

$$\forall i \in I, \forall \mathbf{x} \in \mathbb{F}_p^n. \begin{cases} x_i := \mathfrak{a}_i^\sigma(\mathbf{x}) & ; \forall \sigma \in \Sigma_{o,i} \\ \mathfrak{g}_i^\sigma(\mathbf{x}) - 1 = 0 & ; \forall \sigma \in \Sigma_{c,i} \end{cases} \quad (5)$$

III. Modeling Event-Transferring Communication in DSDES framework

A. General Considerations

In DSDES framework, communication among decentralized supervisors is needed for reevaluation of their guard and updating functions. For example assume that the vector of values after a string s is observed is $\underline{v} := \mathcal{A}(s, \underline{0})$. Then $\sigma \in \Sigma_i$ is enabled at s if and only if $\underline{v} \in \mathcal{G}_i(\sigma)$. To determine if this is the case, $\mathbf{S}_i$ may need to receive the value v_j, for some $j \neq i$, from $\mathbf{S}_j$. When σ is taken, $\mathbf{S}_i$ updates v_i with the value $\mathcal{A}_i(\sigma, \underline{v})$. Again, to correctly evaluate $\mathcal{A}_i(\sigma, \underline{v})$, $\mathbf{S}_i$ may need to receive the value v_j, for some $j \neq i$, from $\mathbf{S}_j$. In PDS (5), $\mathbf{S}_i$'s *informational dependency* is reflected in the functional forms of its updating and guard polynomials and their dependency on external variables. Such observations form the basis to model the communication, define the communication problem, and synthesize solutions for it.

Assume PDS representation (5) is given over $\mathbb{F}_p^n$. Let the network of n supervisors be strongly connected, data transfer be instant with no loss, and disabling controllable events affects none of communication-related events (to keep the network connected). Denote a communication-related event which is issued by $\mathbf{S}_j$ to transfer information to $\mathbf{S}_i$ by subscript indices ji, and assume that it is observable by both supervisors and controllable by $\mathbf{S}_j$. Although $\Sigma_{o,j}$, $\Sigma_{c,j}$, and $\Sigma_{o,i}$ should be enlarged by these new events, to keep the notation simpler, we avoid introducing new sets and assume that this fact is clear from the context.

The event-driven nature of DESs limits the release of the system-related information to the occurrence of observable events such that only the supervisors, who observe them, gain information about the system's evolution. As an information-providing mean, communication should rely on the observation of the occurrence of events, i.e. issuing a communication event would follow the occurrence of an observable event, only. The semantics of each communication event is defined by a map from a subset of observable events to an information-containing set. The communication problem then reduces to designing these maps.

For a DSDES, the system-related information is captured by $\mathbf{x}$, which is, in turn, updated upon the occurrence of observable events. Therefore, the information transferred by a communication event, i.e. the elements of its image set, should be taken from (bit-wise encodings of) either $\mathbf{x}$ or Σ_o. Here, we focus on the latter and refer to it as *event-transferring communication*. To make it precise, first we define a Boolean encoding for observable events.

Definition 2 Associated with PDS (5), let $e = \lceil log_2(\max_{i \in I} |\Sigma_{o,i} \cup \{\epsilon\}|) \rceil$, $\mho = \{1, 2, \cdots, e\}$, and $i, j \in I,\ j \neq i$. Denote by $Y_{ii} = \{y_{ii}^k \mid k \in \mho\}$ the set of $\mathbf{S}_i$'s *private event-encoding* Boolean variables and let every event $\sigma \in \Sigma_{o,i} \cup \{\epsilon\}$ be encoded[2] arbitrarily as $\sigma = (y_{ii}^e \cdots y_{ii}^1)$. The extra artificial event, "ϵ," is used to distinguish the case where no event has occurred initially. Denote by y_{ij}^k a copy of $y_{jj}^k \in Y_{jj}$ stored by $\mathbf{S}_i$ and let $Y_{ij} = \{y_{ij}^k \mid y_{jj}^k \in Y_{jj}\}$, $Y_{ci} = \bigcup_{j \in I \setminus \{i\}} Y_{ij}$, $Y_i = Y_{ii} \dot\cup Y_{ci}$, and $Y = \bigcup_{i \in I} Y_i$. Assume that all variables in Y are initially equal to 0. □

B. Communication-related events

We distinguish two types of communication events which are referred to as $\mathscr{I}$ and $\mathscr{R}$ events. This classification divides the communication design into two levels of *information exchange* and *routing*, which, respectively address what information needs to be exchanged mutually among supervisors and how these exchanges can be performed using the available communication channels. We limit the focus of this paper to $\mathscr{I}$ events, introduced next. Let $i, j \in I, i \neq j$.

$\mathscr{I}$ **events**: An event $\mathscr{I}_{ji}$ transfers $\mathbf{S}_j$'s private information to $\mathbf{S}_i$. Let $\Sigma_{\mathscr{I},j} \subseteq (\Sigma_{o,j} \cup \bigcup_{k \in I, k \neq j} \{\mathscr{I}_{kj} \mid \mathscr{I}_{kj} \text{ is defined}\})$ denote the set of observable events by $\mathbf{S}_j$, after which $\mathbf{S}_j$ issues an $\mathscr{I}$ event. Correspondingly, define $\mathscr{I}_{ji} : \Sigma_{\mathscr{I},j} \to pwr(Y_{jj})$ as a function which associates to an event in $\Sigma_{\mathscr{I},j}$, a piece of information stored by event-encoding Boolean variables in Y_{jj} according to a rule which will become specified upon the design of the communication. Whereas the definition of $\Sigma_{\mathscr{I},j}$ allows the firing of an $\mathscr{I}$ event after another $\mathscr{I}$ event, circular definitions should be avoided. Since I is finite, there is a finite number of distinct $\mathscr{I}$ events and this, together with the finiteness of Y, guarantee that the information is exchanged in a finite number of communication steps. Once received by $\mathbf{S}_i$, $\mathscr{I}_{ji}(.)$ provides it with the updated copies of the variables in its image, i.e.

$$\forall i, j \in I, i \neq j,\ \forall k \in J,\ \forall y_{jj}^k \in Y_{jj},$$
$$\forall \sigma \in \Sigma_{\mathscr{I},j}. \quad y_{jj}^k \in \mathscr{I}_{ji}(\sigma) \Longrightarrow y_{ij}^k := y_{jj}^k. \quad (6)$$

In a strongly connected network, $\mathscr{I}$ events can singly implement the exchange of information. However, if some direct communication links are missing or due to channel constraints "indirect" data transfers are preferred, $\mathscr{R}$ events should be designed. This issue is left for future work. Notice that even in the second case, $\mathscr{I}$ events still specify what information needs to exchange between supervisors.

Inherently, the definition of $\mathscr{I}_{ji}(\sigma)$ determine *who* (i.e. $\mathbf{S}_j$) sends *what* (a subset of Y_{jj}) to *whom* (i.e. $\mathbf{S}_i$). Implicitly, the dependency on an event as its argument, bears a notion of "logical" time which roughly specifies the soonest moment at which the communication can start. Also notice that here communication is *event-triggered* and its content, being a (Boolean)-variable representation of the events, is *event-based*. An event-based "communication problem" can be defined within DSDES framework as follows.

Definition 3 Let PDS (5) and its event-encoding in Definition 2 be given. An *information policy* for (5) is equivalent to designing $(\Sigma_{\mathscr{I},j}, \mathscr{I}_{ji}(.))$ for every $i, j \in I$. □

[2]Although some $\Sigma_{o,i}$s may have less than $e-1$ elements, we choose a common e for the sake of notational simplicity. Clearly, if $|\Sigma_{o,i}| < e-1$, some higher significant bits in Y_{ii} would be constantly equal to 0.

Problem 2 *Event-Transferring Communication Problem in DSDES framework:* Associated with Problem 1 and Proposition 1, let PDS (5) represent the DSDES and the event-encoding in Definition 2 be used. Find an information policy such that $L(\mathcal{D}) = \overline{E}$ and $L_m(\mathcal{D}) = E$. □

Communication is the third mean, on top of observation and control, with which decentralized supervisors confine a plant's behavior within given specifications. By presenting the system information of the centralized supervisor in a distributed way and putting it in PDS form, DSDES framework provides a general, flexible, and systematic approach for analysis and synthesis of decentralized supervisors. Supervisors' private variables form the largest set of system information, owned by a given state representation of the centralized supervisor. Event-transferring communication helps each supervisor reevaluate its guard and updating functions by providing to it the last observed events which affect the external variables on which it depends. Computation of communication, as a solution to Problem 2, deserves a separate work of its own, which is based on the study of algebraic structures. Here, we illustrate the applicability of the proposed approach by designing an information policy and verifying its correctness.

IV. Synthesis of Event-Transferring Communication in DSDES framework

A. Informational Dependency of Supervisors

As stated before, a supervisor's dependency on external variables calls for a communication which provides the required information to the supervisor to make itself updated. In this subsection, we formalize the informational dependency for each supervisor.

Definition 4 For every $i \in I$, define $\mathcal{N}_{\mathfrak{a}_i} = \{j \in I \mid \exists\sigma \in \Sigma_{o,i}.\ x_j \in \arg(\mathfrak{a}_i^\sigma)\}$ and $\mathcal{N}_i = \{j \in I \mid [\exists\sigma \in \Sigma_{o,i}.\ x_j \in \arg(\mathfrak{a}_i^\sigma)] \vee [\exists\sigma' \in \Sigma_{c,i}.\ x_j \in \arg(\mathfrak{g}_i^{\sigma'})]\}$. Define also set $\mathcal{O}_i \subseteq I$ recursively as follows.

$\mathcal{N}_i \subseteq \mathcal{O}_i \quad\wedge\quad [\forall j \in I.\ j \in \mathcal{O}_i \implies \forall k \in \mathcal{N}_{\mathfrak{a}_j}.\ k \in \mathcal{O}_i]$ □

Lemma 1 We have the following.

$\forall i \in I, \forall j \in \mathcal{O}_i, \forall\sigma \in \Sigma_{o,j}.\ x_k \in \arg(\mathfrak{a}_j^\sigma) \implies k \in \mathcal{O}_i$ ■

In simple words, what Definition 4 means is that $\mathcal{O}_i$ is the largest set of $\mathbf{S}_j$s, on whose private information, i.e. x_js, $\mathbf{S}_i$ depends, where $i, j \in I$. This dependency is either explicit, i.e. when x_j appears as the argument of an updating or a guard function of $\mathbf{S}_i$, or implicit, i.e. when the updating function corresponding to such an $x_j, j \neq i$, depends on $x_k, k \neq j, i$.

Lemma 2 For every $i \in I$, $\mathcal{O}_i$ is a fixed point of function $F : \mathscr{P}(I) \to \mathscr{P}(I) : A \mapsto \left(A \cup \bigcup_{j\in A} \mathcal{N}_{\mathfrak{a}_j}\right)$. ■

Clearly F is monotonic and the finiteness of I implies that for every $\mathcal{N}_i \subseteq I$, $F(\mathcal{N}_i)$ converges to fixed point $\mathcal{O}_i$. This can be regarded as the basis of an algorithm to compute $\mathcal{O}_i$.

Algorithm 1 *Computation of $\mathcal{O}_i$:* Associated with PDS (5), for every $i \in I$ do the following.

1) Compute $\mathcal{N}_i = \{j \in I \mid [\exists\sigma \in \Sigma_{o,i}.\ x_j \in \arg(\mathfrak{a}_i^\sigma)] \vee [\exists\sigma' \in \Sigma_{c,i}.\ x_j \in \arg(\mathfrak{g}_i^{\sigma'})]\}$.
2) Set $A = \mathcal{N}_i$ and $B = F(A)$.
3) While $A \neq B$ do $[A := B$ and $B := F(A)]$.
4) Report $\mathcal{O}_i = A$. □

Following the concept of $\mathcal{O}_i$, if supervisor $\mathbf{S}_i$ stores a copy of the private variables it requires (directly or indirectly) to compute its guard and updating functions together with the copies of updating functions which are used to reevaluate those copy variables, it can independently compute the new values of the copy variables if it is informed of what observable events occur. The next definition makes it clear what we mean by copy variables and updating equations.

Definition 5 Associated with PDS (5) and for each $i \in I$ and every $j \in \mathcal{O}_i \setminus \{i\}$, let x_j^i be the copy of x_j which is stored by $\mathbf{S}_i$. Let $\mathbf{x}^i$ be the vector of all x_j^is such that they are sorted from left to right based on their index j (i.e. in the same order they appear in $\mathbf{x}$). Write $\mathbf{x}^i = \mathbf{x}$ to state $\forall i \in I, \forall j \in \mathcal{O}_i \setminus \{i\}.\ x_j^i = x_j$. Also assume that all x_j^is are initialized to 0. Correspondingly, denote by $\mathfrak{a}_{ij}^\sigma(\mathbf{x}^i)$ the copy of updating function $\mathfrak{a}_j^\sigma(\mathbf{x})$ stored by $\mathbf{S}_i$. □

Lemma 3 For every $i \in I$, $j \in \mathcal{O}_i \setminus \{i\}$, and $\sigma \in \Sigma_{o,j}$, $\mathfrak{a}_{ij}^\sigma(\mathbf{x}^i)$ in Definition 5 is well defined. ■

To be able to compute copies of other supervisors' private variables, each $\mathbf{S}_i$ should store a copy of their updating functions. We take this point for granted in the following.

Assumption 1 Assume that for each $i \in I$, each $j \in \mathcal{O}_i \setminus \{i\}$, and each $\sigma \in \Sigma_{o,j}$, $\mathbf{S}_i$ stores $\mathfrak{a}_{ij}^\sigma(\mathbf{x}^i)$, i.e. it stores a copy of each of $\mathbf{S}_j$'s *non-identity updating functions.* □

Under Assumption 1, each supervisor $\mathbf{S}_i$ can independently compute the new values of the private variables of other $\mathbf{S}_j$s upon being informed of the occurrence of every observable event by those $\mathbf{S}_j$s. This paves the way to define an information policy which communicates the observable events, rather than state-based information. As Definition 2 reads, every supervisor assigns a code to each of its observable events. To decode these codes upon their arrival, the target supervisor should have a look up table which stores the codes. This is stated in the following assumption. Notice that for each $\sigma \in \Sigma_o$, every supervisor whose index is in $I_o(\sigma)$, assigns its own code to σ.

Assumption 2 For each $i \in I$, each $j \in \mathcal{O}_i$, and every $\sigma \in \Sigma_{o,j}$, $\mathbf{S}_i$ has a look up table which stores the code(s) for σ (as assigned by different supervisors whose index is in $I_o(\sigma)$). □

Once a supervisor observes the occurrence of an event, on top of updating its own private variable, it updates copies of all other private variables which it keeps and are affected by that event, as summarized in the following assumption.

Assumption 3 For every $i \in I$, each $j \in \mathcal{O}_i$, and each $\sigma \in \Sigma_{o,i} \cap \Sigma_{o,j}$, if σ occurs and $\mathbf{S}_i$ is informed of this occurrence, $\mathbf{S}_i$ computes $\hat{x}_j^i := \mathfrak{a}_{ij}^\sigma(\mathbf{x}^i)$. □

B. An event-transferring information policy

We have now enough means to introduce a solution to Problem 2. The following definition will be helpful in defining the new policy.

Definition 6 For each $\sigma \in \Sigma_o$ define $I_{com}(\sigma) = k$, where $k \in I_o(\sigma)$. □

Notice that for each $\sigma \in \Sigma_o$, $I_o(\sigma) \neq \emptyset$, and thus $I_{com}(\sigma)$ is well defined.

Definition 7 Considering PDS (5) and following Definitions 5 and 2, for each $i \in I$, let $\alpha_i \in \Sigma_{o,i} \cup \{\epsilon\}$, encoded as $\alpha_i = (y^e_{ii}, \cdots, y^1_{ii})$, be the last event observed by $\mathbf{S}_i$. Assume that for each $\sigma \in \Sigma_o$, $I_{com}(\sigma)$ is given as in Definition 6. For PDS (5) *information policy* $\mathscr{E}$ is as follows:
For all $i, j \in I$ such that $i \neq j$ we have $\Sigma_{\mathscr{I},j} = \Sigma_{o,j}$, and

$$\forall \sigma \in \Sigma_{o,j}.\ \mathscr{I}_{ji}(\sigma) = \{\hat{y}^k_{jj} \in Y_{jj} |\ \sigma = (\hat{y}^e_{jj}, \cdots, \hat{y}^1_{jj}) \wedge \hat{y}^k_{jj} \neq y^k_{jj} \wedge I_{com}(\sigma) = j \wedge i \notin I_o(\sigma) \wedge [\exists l \in \mathcal{O}_i \backslash \{i\}.\ l \in I_o(\sigma)]\} \quad (7)$$

Once $\mathscr{I}_{ji}$ is received by $\mathbf{S}_i$, the copies of event-encoding Boolean variables will be updated, i.e.

$$\forall i, j \in I, i \neq j, \forall k \in \mho, \forall \hat{y}^k_{jj} \in Y_{jj}, \forall \sigma \in \Sigma_{\mathscr{I},j}.$$
$$(1)\ [\hat{y}^k_{jj} \in \mathscr{I}_{ji}(\sigma) \Longrightarrow \hat{y}^k_{ij} := \hat{y}^k_{jj}] \wedge [\hat{y}^k_{jj} \notin \mathscr{I}_{ji}(\sigma) \Longrightarrow \hat{y}^k_{ij} := y^k_{ij}]$$
$$\wedge (2)\ [\forall \beta \in \Sigma_{o,j}, \forall m \in \mathcal{O}_i \backslash \{i\}, \forall \mathbf{x}^i \in \mathbb{F}_p^{|\mathcal{O}_i|}.\ (\hat{y}^e_{ij}, \cdots, \hat{y}^1_{ij}) = \beta \wedge m \in I_o(\beta) \Longrightarrow \hat{x}^i_m := \mathfrak{a}^\beta_{im}(\mathbf{x}^i)] \quad \square$$

In simple words, upon the occurrence of $\sigma \in \Sigma_o$, information policy $\mathscr{E}$ requires that one of the supervisors, that is the one specified by $I_{com}(\sigma)$ (which observes the occurrence of σ), informs the supervisors which cannot observe σ and whose guards or updating functions are affected by its occurrence directly or indirectly, of this occurrence, as reflected in (7). As (7) reads, when the sent bits arrive at the target supervisor(s), first they are decoded by the receivers to find out which event has been observed by the sender (Conjunct 1), and then all copies of the private variables, which are kept by each receiver and are affected by the decoded event, are updated using the copies of the corresponding updating functions (Conjunct 2).

Proposition 3 *A solution to Problem 2:* Associated with Problem 2 and under Assumptions 1, 2, and 3, if the network of supervisors is strongly connected with lossless channels and if communication is instantaneous, information policy $\mathscr{E}$ insures that $L(\mathcal{D}) = \overline{E}$ and $L_m(\mathcal{D}) = E$. ■

Definition 7 assumes a given $I_{com}(\sigma)$ for every $\sigma \in \Sigma_o$ and introduces information policy $\mathscr{E}$ based on it. As (7) reads, $\mathbf{S}_{I_{com}(\sigma)}$ is responsible for issuing communication event $\mathscr{I}_{I_{com}(\sigma)i}$ to all $\mathbf{S}_i$s which cannot observe σ, but have a copy variable which can be affected by σ. Following Definition 6, when $I_o(\sigma)$ is a singleton, there is just one choice for $I_{com}(\sigma)$. However, if $|I_o(\sigma)| > 1$, there are different supervisors which can be the issuer of $\mathscr{I}_{I_{com}(\sigma)i}$. By Conjunct 2 of (7), once the communication is received, regardless of which supervisor (among those whose index is in $I_o(\sigma)$) has issued it, it results in the same process, i.e. reevaluation of all copy variables, stored by $\mathbf{S}_i$, which are affected by σ. However, by (7) and Conjunct 1 of (7), each choice of $I_{com}(\sigma)$ would entail communication of the changed bits of $Y_{I_{com}(\sigma)I_{com}(\sigma)}$. Therefore, it is plausible to ask if there are choices of $I_{com}(\sigma)$s which lead to a minimal communication. Whereas this issue is not formally investigated here, it is worth mentioning, as a rule of thumb, that if a supervisor $\mathbf{S}_l$ has fewer number of observable events, it has probably more constant bits in Y_{ll}. Therefore, choosing $I_{com}(\sigma) = l$ can reduce the content of $\mathscr{I}_{li}(\sigma)$. Furthermore, for $l, i \in I$ and $\sigma \in \Sigma_{o,l}$ such that $I_{com}(\sigma) = l$, if the employed event-encoding scheme for $\mathbf{S}_l$ is such that along the system's evolution, fewer bits in Y_{ll} change with the occurrence of an event σ, the content of $\mathscr{I}_{li}(\sigma)$ would be decreased, too. The next example illustrates the procedure.

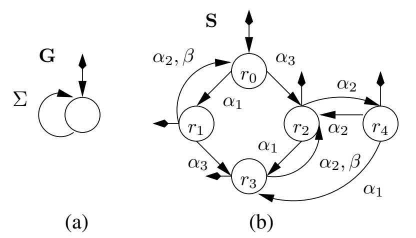

Fig. 1. (a) The plant's model. (b) The specification which is also a centralized supervisor.

TABLE I
UPDATING FUNCTIONS ($a, b \in \{0,1,2\}$, $c \in \{0,1\}$)

$\underline{v}$	$00c$	$21c$	$12c$	201	111	021	else
$\mathcal{A}(\alpha_1, \underline{v})$	$10c$	$01c$	$22c$	101	011	221	$-$
$\underline{v}$	$01c$	$10c$	$22c$	021	111	201	001
$\mathcal{A}(\alpha_2, \underline{v})$	$00c$	$12c$	$21c$	001	121	211	021
$\underline{v}$	121	211	else				
$\mathcal{A}(\alpha_2, \underline{v})$	111	201	$-$				
$\underline{v}$	$10c$	$01c$	$22c$	else			
$\mathcal{A}(\beta, \underline{v})$	$21c$	$12c$	$00c$	$-$			
$\underline{v}$	$ab0$	else					
$\mathcal{A}(\alpha_3, \underline{v})$	$ab1$	$-$					

Example 1 Figure 1-a shows the model of a distributed network consisting of three plant components and four events $\alpha_1, \alpha_2, \alpha_3$ and β, where $\Sigma_{c,1} = \Sigma_{o,1} = \{\alpha_1, \beta\}$, $\Sigma_{c,2} = \Sigma_{o,2} = \{\alpha_2, \beta\}$, and $\Sigma_{c,3} = \Sigma_{o,3} = \{\alpha_3\}$. The specification $\mathbf{S}$ is shown in part (b) and all its states are marked. Observe that $\mathbf{S}$ is a proper centralized supervisor enforcing itself. An ALM can be defined for $\mathbf{S}$ as follows (A point $(a, b, c) \in \mathbb{N}^3$ is denoted by 'abc') $\ell(r_0) = \{000, 210, 120\}$, $\ell(r_1) = \{100, 010, 220\}$, $\ell(r_2) = \{001, 211, 121\}$, $\ell(r_3) = \{101, 011, 221\}$, and $\ell(r_4) = \{201, 021, 111\}$. Correspondingly, guard functions are computed as follows: $\mathcal{G}_1(\alpha_1) = \ell(r_0) \cup \ell(r_2) \cup \ell(r_4)$, $\mathcal{G}_2(\alpha_2) = \ell(r_1) \cup \ell(r_2) \cup \ell(r_3) \cup \ell(r_4)$, $\mathcal{G}_1(\beta) = \mathcal{G}_2(\beta) = \ell(r_1) \cup \ell(r_3)$, $\mathcal{G}_3(\alpha_3) = \ell(r_0) \cup \ell(r_1)$. The updating functions are listed in Table I. Arbitrary cases are denoted by "$-$" and are used to simplify the guard and updating functions. Table II shows the polynomials which represent the updating and guard functions computed using Algorithm 1 in [11]. Here $\mathbf{x} = [x_1, x_2, x_3]'$, $V_1 = V_2 = \{1, 2, 3\}$, $V_3 = \{1, 2\}$, $p_1 = p_2 = 3$, and $p_3 = 2$. Step 1 of Algorithm 1 in [11] yields $p = 3$, i.e. $\mathbb{F}_3$ is the underlying field.

Let us apply information policy $\mathscr{E}$ for the DSDES. To this end, from Table II we have following.

$$\mathcal{N}_{\mathfrak{a}_1} = \mathcal{N}_{\mathfrak{a}_2} = \{1, 2\},\ \mathcal{N}_{\mathfrak{a}_3} = \{3\},$$
$$\mathcal{N}_1 = \{1, 2\},\ \mathcal{N}_2 = \{1, 2, 3\},\ \mathcal{N}_3 = \{3\}$$

Application of Algorithm 1 then leads to $\mathcal{O}_1 = \{1, 2\}$, $\mathcal{O}_2 = \{1, 2, 3\}$, and $\mathcal{O}_3 = \{3\}$.

TABLE II

COMPUTED POLYNOMIALS FOR UPDATING AND GUARD FUNCTIONS

$x_1 := \mathfrak{a}_1^{\alpha_1}(\mathbf{x}) = 2(x_2+2), \quad x_1 := \mathfrak{a}_1^{\beta}(\mathbf{x}) = x_1 + 1$
$x_2 := \mathfrak{a}_2^{\alpha_2}(\mathbf{x}) = x_1(x_1+1) + (x_2+2)(2x_1+x_2+1)$
$x_2 := \mathfrak{a}_2^{\beta}(\mathbf{x}) = x_2 + 1, \qquad x_3 := \mathfrak{a}_3^{\alpha_3}(\mathbf{x}) = 2(x_3^2+2)$
$\mathfrak{g}_1^{\alpha_1}(\mathbf{x}) = 2(x_1+2)^2(x_2^2+2) + 2x_1^2x_2(x_2+1) + 2(x_1+1)^2x_2(x_2+2)$
$\mathfrak{g}_2^{\alpha_2}(\mathbf{x}) = 2x_1^2(x_2^2+2) + 2(x_1+1)^2x_2(x_2+1) + 2(x_1+2)^2x_2(x_2+2) + 2x_3(x_3+1)[x_1x_2 + 2x_1^2 + 2x_2^2 + 1]$
$\mathfrak{g}_{1,2}^{\beta}(\mathbf{x}) = x_1(x_1+1)(x_2^2+2) + (x_1^2+2)x_2(x_2+1) + x_1(x_1+2)x_2(x_2+2)$
$\mathfrak{g}_3^{\alpha_3}(\mathbf{x}) = 2(x_3^2+2)$

TABLE III

ENCODING OF EVENTS FOR EXAMPLE 1

$\mathbf{S}_1 : (y_{11}^2, y_{11}^1)$	$\epsilon = (00)$	$\alpha_1 = (01)$	$\beta = (11)$
$\mathbf{S}_2 : (y_{22}^2, y_{22}^1)$	$\epsilon = (00)$	$\alpha_2 = (01)$	$\beta = (11)$
$\mathbf{S}_3 : (y_{33}^2, y_{33}^1)$	$\epsilon = (00)$	$\alpha_3 = (01)$	

Observe that $|\Sigma_{o,1}| = |\Sigma_{o,2}| = 2$ and $|\Sigma_{o,3}| = 1$, therefore following Definition 2 we have $e = \lceil log_2(2+1) \rceil = 2$. Also it holds that $I_o(\alpha_1) = \{1\}, I_o(\alpha_2) = \{2\}, I_o(\beta) = \{1,2\}$, and $I_o(\alpha_3) = \{3\}$. Following Definition 6 we have $I_{com}(\alpha_1) = 1, I_{com}(\alpha_2) = 2$, and $I_{com}(\alpha_3) = 3$. However, $I_{com}(\beta)$ can be either 1 or 2 and, based on the fact that $\mathbf{S}_1$ and $\mathbf{S}_2$ has each two observable events, there does not seem to be any difference between choosing either supervisors, hence we set $I_{com}(\beta) = 1$.

Codes for the events are shown in Table III. Here, all supervisors encode ϵ as (00), and we have $\alpha_1 = (01), \alpha_2 = (01)$, and $\alpha_3 = (01)$ [3]. To encode β, we notice that if $\beta = (10)$, then between $\alpha_1 = (01)$ and $\beta = (10)$ there would be a difference of 2 bits, i.e. upon the occurrence of one of these events after the other one, $\mathbf{S}_1$ should send 2 bits. However, if $\beta = (11)$, the difference would be between (01) and (11), i.e. just 1 bit. Therefore, this latter choice is taken as the code of β assigned by $\mathbf{S}_1$. Although, $\mathbf{S}_2$ needs not inform others of the occurrence of β, it encodes β as (11), because (10) would require $\mathbf{S}_2$ to send 2 bits when reporting the occurrence of α_2 after β. Moreover, a common code for β can save some memory space, by reducing from two different spaces to one in the look up table of the events, and helps supervisors distinguish β consistently in the case of communication faults.

Using the above codes and Definition 7, communication events can be computed. To this end, assume that the last event observed by $\mathbf{S}_i$ is $\sigma_i \in \Sigma_{o,i} \cup \{\epsilon\}$, encoded as (y_{ii}^2, y_{ii}^1) and the new event is encoded as $(\hat{y}_{ii}^2, \hat{y}_{ii}^1)$, where $i \in \{1,2,3\}$. The communication events are shown in Table IV, where conditions of selecting communication content are simply formulated as whether corresponding $\hat{y}$s and ys are equal to each other or not. For $\mathbf{S}_3$, it can be seen that $y_{33}^2 = 0$ holds all the time, hence simplifying the content condition. $\Diamond$

[3]Notice that these codes are assigned to different variables (y_{ii}^2, y_{ii}^1) for $i = 1, 2$, and 3, respectively, as shown in Table III.

TABLE IV

COMMUNICATION AMONGST SUPERVISORS IN EXAMPLE 1 BASED ON POLICY $\mathscr{E}$

$\mathscr{I}_{12}(\alpha_1) = \begin{cases} \{\hat{y}_{11}^1, \hat{y}_{11}^2\} & ;\text{if } (\hat{y}_{11}^1 \neq y_{11}^1) \wedge (\hat{y}_{11}^2 \neq y_{11}^2) \\ \{\hat{y}_{11}^1\} & ;\text{if } (\hat{y}_{11}^1 \neq y_{11}^1) \wedge (\hat{y}_{11}^2 = y_{11}^2) \\ \{\hat{y}_{11}^2\} & ;\text{if } (\hat{y}_{11}^1 = y_{11}^1) \wedge (\hat{y}_{11}^2 \neq y_{11}^2) \\ \emptyset & ;\text{if } (\hat{y}_{11}^1 = y_{11}^1) \wedge (\hat{y}_{11}^2 = y_{11}^2) \end{cases}$
$\mathscr{I}_{13}(\alpha_1) = \mathscr{I}_{13}(\beta) = \mathscr{I}_{12}(\beta) = \emptyset$
$\mathscr{I}_{21}(\alpha_2) = \begin{cases} \{\hat{y}_{22}^1, \hat{y}_{22}^2\} & ;\text{if } (\hat{y}_{22}^1 \neq y_{22}^1) \wedge (\hat{y}_{22}^2 \neq y_{22}^2) \\ \{\hat{y}_{22}^1\} & ;\text{if } (\hat{y}_{22}^1 \neq y_{22}^1) \wedge (\hat{y}_{22}^2 = y_{22}^2) \\ \{\hat{y}_{22}^2\} & ;\text{if } (\hat{y}_{22}^1 = y_{22}^1) \wedge (\hat{y}_{22}^2 \neq y_{22}^2) \\ \emptyset & ;\text{if } (\hat{y}_{22}^1 = y_{22}^1) \wedge (\hat{y}_{22}^2 = y_{22}^2) \end{cases}$
$\mathscr{I}_{23}(\alpha_2) = \mathscr{I}_{23}(\beta) = \mathscr{I}_{21}(\beta) = \emptyset$
$\mathscr{I}_{32}(\alpha_3) = \begin{cases} \{\hat{y}_{33}^1\} & ;\text{if } (\hat{y}_{33}^1 \neq y_{33}^1) \\ \emptyset & ;\text{if } (\hat{y}_{33}^1 = y_{33}^1) \end{cases}$
$\mathscr{I}_{31}(\alpha_3) = \emptyset$

It is worth mentioning again that the emphasis here is not on the solution approaches and their computational efficiency, but rather on illustrating the modeling and synthesis capabilities of DSDES framework.

REFERENCES

[1] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control and Optimization*, vol. 25, no. 1, pp. 206–230, Jan. 1987.

[2] K. Rudie and W. M. Wonham, "Think globally, act locally: Decentralized supervisory control," *IEEE Trans. Automat. Contr.*, vol. 37, pp. 1692–1708, Nov. 1992.

[3] J. H. van Schuppen, "Decentralized control with communication between controllers," in *Unsolved Problems in Mathematical Systems and Control Theory*, V. D. Blondel and A. Megretski, Eds. Princeton, USA: Princeton University Press, 2004.

[4] D. Teneketzis, "On information structures and nonsequential stochastic control," *CWI Quarterly*, vol. 10, no. 2, pp. 179–199, 1997.

[5] A. Mannani and P. Gohari, "A framework for modeling communication among decentralized supervisors for discrete-event systems," in *Proc. of IEEE Conference on Sys., Man, and Cyb. (SMC'07)*, Montreal, Canada, Oct. 2007, pp. 1339–1344.

[6] ——, "Decentralized supervisory control of discrete-event systems over communication networks," *IEEE Trans. on Automat. Contr.*, vol. 53, no. 2, pp. 547–559, Mar. 2008.

[7] M. L. Borgne, A. Benveniste, and P. L. Guernic, "Polynomial dynamical systems over finite fields," in *Algebraic Computing in Control*. Heidelberg: Springer Berlin, 1991, vol. 165/1991, pp. 212–222.

[8] J. Gunnarsson, "Symbolic methods and tools for discrete event dynamic systems," Ph.D. dissertation, Univ. of Linköping, S-581 83 Linköping, Sweden.

[9] K. Åkesson, M. Fabian, H. Flordal, and R. Malik, "Supremica - an integrated environment for verification, synthesis and simulation of discrete event systems," in *Proc. IEEE 8th International Workshop on Discrete Event Systems (WODES'06)*, Ann Arbor, MI, USA, July 2006, pp. 384 – 385.

[10] C. Ma and W. M. Wonham, "Nonblocking supervisory control of state tree structures," *IEEE Trans. on Automat. Contr.*, vol. 51, no. 5, pp. 782–793, May 2006.

[11] A. Mannani and P. Gohari, "Formal modeling and synthesis of state-transferring communication among decentralized supervisors for discrete-event systems," in *Proc. IEEE Conf. on Systems, Man, and Cybernetics SMC*, San Antonio, TX, Oct. 2009.

[12] R. Germundsson, "Symbolic systems: Theory, computation, and applications," Ph.D. dissertation, Univ. of Linköping, S-581 83 Linköping, Sweden, Sept. 1995. [Online]. Available: http://www.control.ee.liu.se

[13] R. Lidl and H. Niederreiter, *Finite Fields*, 1st ed. Cambridge University Press, 1997.