Abstractions for nonblocking supervisory control of Extended Finite Automata

Abstractions for Nonblocking Supervisory Control of Extended Finite Automata

Mohammad Reza Shoaei, Lei Feng, Bengt Lennartson

***Abstract*—An abstraction method for Extended Finite Automata (EFAs), i.e., finite automata extended with variables, using transition projection is presented in this work. A manufacturing system modeled by EFAs is abstracted into subsystems that embody internal interacting dependencies. Synthesis and verification of subsystems are achieved through their model abstractions rather than their global model. Sufficient conditions are presented to guarantee that supervisors result in maximally permissive and nonblocking control. An examples demonstrate the computational effectiveness and practical usage of the approach.**

I. Introduction

Increasing global competition is changing the strategies and methods within industries in order to maintain market presence and boost growth in an increasingly tough competitive environment. The industries are pushed to enhance performance and reduce the time and cost of introducing new products. One contribution is to automatically compute the controllers that coordinate their manufacturing plant. Indeed, such controllers must guarantee the safety and nonblocking of the controlled system.

Supervisory Control Theory (SCT), established by Ramadge and Wonham [1], is a formal framework for modeling and control of *Discrete-Event Systems* (DES). Application domains include manufacturing systems, vehicular traffic, robotics, computer, and communication networks. Problems that SCT can address include dynamic allocation of resources, the prevention of system blocking, etc. and, within these constraints, maximally permissive system behavior. SCT can systematically synthesize supervisory controllers that are able to prevent a DES from executing undesirable behavior. Nevertheless, industrial acceptance is still scarce. A number of issues that hinder industrial use have been identified in [2], [3]. Main issues are the discrepancy between the signal based reality and the event-based automata framework, the lack of a compact representation of large models, and computational complexity.

Sköldstam et al. [4] tackle these issues by introducing a modeling formalism, called Extended Finite Automata (EFAs), which are ordinary automata extended with discrete variables, guard expressions and action functions. This modeling formalism has been used in several research works such as [5], [6], including different approaches for synthesizing EFAs [7], [8] and an implementation in the DES tool Supremica [9].

Although EFAs ease the modeling by providing a compact representation of the system, SCT analysis is performed on their underlying automata models and therefore, the fundamental obstruction to the development of SCT, i.e., the computational complexity of synthesizing nonblocking supervisors, still remains. Indeed, the nonblocking supervisory control problem for DES is NP-hard [10]. The exponential complexity of supervisor design arises from synchronizing components into a global model. Therefore, effective control methods for various subclasses of DES that enjoy special structures are introduced such as *modularity* [11], [12] and *model abstraction* [13], [14].

The most effective model abstraction operator in SCT is the causal reporter map having the observer property [15], which later was substituted by Feng and Wonham [16], [17] with the natural observer. In this, components which are sharing only a small number of common events, their abstractions tend to be small, and designing controllers may require only a modest effort. A limitation of the natural projection is the need for the language of a system to be known or be obtained by its generators, for instance, automata which is not the case for a DES modeled by EFAs. The transitions in EFAs are augmented with guards and actions which one cannot define the language of individual components without considering the global behavior of the system, i.e., the synchronous composition of all components.

In this paper, we substitute the natural projection with transition projection to achieve model abstraction for EFAs. To reduce the computational complexity, the controller is synthesized based on the model abstraction of subsystems rather than the global system. Sufficient conditions are presented to guarantee the decentralized supervisors to result in maximally permissive and nonblocking control of the entire system.

The paper is organized as follows: Section II briefly describes the EFA and its properties. In Section III, a model abstraction using transition projection is introduced followed by Section IV in which the observer property for transition projections is explained. An example is presented in Section V to demonstrate the practical usage of the method. Finally, Section VI concludes our work.

M.R. Shoaei, B. Lennartson are with Department of Signals and Systems, Chalmers University of Technology, SE-412 96, Gothenburg, Sweden, {shoaei, bengt.lennartson}@chalmers.se, L. Feng is with the Department of Machine Design, KTH - Royal Institute of Technology, SE-100 44 Stockholm, Sweden, leifeng@md.kth.se. This work was carried out at the Wingquist Laboratory VINN Excellence Center within the Area of Advance – Production at Chalmers, supported by the Swedish Governmental Agency for Innovation Systems (VINNOVA). The support is gratefully acknowledged.

II. PRELIMINARIES

A. Languages and Automata

The behavior of DES [19], [20] is described in terms of event sequences and regular languages [1]. A regular language is a subset of strings that can be recognized by a *finite automaton* (FA) $G = (Q, \Sigma, \mapsto, Q^0, Q^m)$. Q is the finite *state* set. Σ is a non-empty finite event set called *alphabet*. $\mapsto \subseteq Q \times \Sigma \times Q$ is the state *transition function* mapping elements of $Q \times \Sigma$ into singletons of Q. The element $Q^0 \subseteq Q$ is the set of *initial states* and $Q^m \subseteq Q$ is the set of *marked states*.

The transition relation in G is written in infix notation $p \overset{\sigma}{\mapsto} q$. Let Σ^* be the set of all finite strings over Σ, including the empty string ε. Then, these notations can be extended to strings in Σ^* in the natural way by letting $p \overset{\varepsilon}{\mapsto} p$ for all $p \in Q$ and $p \overset{s\sigma}{\mapsto} q$ if $p \overset{s}{\mapsto} r$ and $r \overset{\sigma}{\mapsto} q$ for some $q, r \in Q, s \in \Sigma^*$, and $\sigma \in \Sigma$. Let $p \overset{\sigma}{\mapsto}$ denote the existence of one state $q \in Q$ such that $p \overset{\sigma}{\mapsto} q$, and $p \mapsto q$ the existence of a string $s \in \Sigma^*$ such that $p \overset{s}{\mapsto} q$. Automaton G is deterministic if Q^0 is a singleton q_0 and $p \overset{\sigma}{\mapsto} q$ and $p \overset{\sigma}{\mapsto} \acute{q}$ always implies $q = \acute{q}$.

Note that, by definition, the symbol ε does not belong to Σ. If it is to be included, the event set $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ is used instead. An important property of an automaton is *nonblocking*. The automaton G is nonblocking if any state reachable from the initial state $q_0 \in Q^0$ can also reach a marked state via some string, i.e., $(\forall q \in Q) q_0 \mapsto q \Rightarrow q \mapsto p$ for some $p \in Q^m$.

Given two event sets Σ and $\Sigma_0 \subseteq \Sigma$, the *natural projection* is the function $P : \Sigma^* \to \Sigma_0^*$ such that $P(\varepsilon) = \varepsilon$ and

$$P(\sigma) = \begin{cases} \varepsilon, & \sigma \in \Sigma - \Sigma_0 \\ \sigma, & \sigma \in \Sigma_0 \end{cases}$$
$$P(s\sigma) = P(s)P(\sigma), s \in \Sigma^*, \sigma \in \Sigma$$

The effect of P on a string $s \in \Sigma^*$ is just to erase the events in s that do not belong to Σ_0, but keep the events in Σ_0 unchanged. The *inverse image* of the natural projection P is a function $P^{-1} : Pwr(\Sigma_0^*) \to Pwr(\Sigma^*)$ where Pwr is the power set.

B. Extended Finite Automata

A finite automaton can be extended with a set of variables to an *Extended Finite Automaton* (EFA) whose transitions are augmented with Boolean conditions and actions on these variables to enjoy a compact and symbolic description of a DES.

Let $\mathcal{V} = \{v_1, \ldots, v_n\}$ be the set of n typed variables and D_i be the domain (type) of v_i. Let η denote a tuple of *variable evaluation* $\eta : \mathcal{V} \to D$ assigning to each variable $v_i \in \mathcal{V}$ its current value in D_i. $\mathcal{G}$ is the set of Boolean conditions over $\mathcal{V}$ in which each condition g, also called guard, is a propositional logic formula whose propositional symbols are of the form $\bar{v} \in \bar{D}$ where $\bar{v} = (v_1, \cdots, v_n)$ is an n-tuple of pairwise distinct variables in $\mathcal{V}$, and $\bar{D}$ is a subset of the domain of variables $D = D_1 \times \cdots \times D_n$. Given two guards g and h, we say that g is a subguard of h, denoted $g \preceq h$, if $g \wedge h = g$, and we say both g and h have the same evaluation for η, denoted $g = h$, if $\eta \models g \Leftrightarrow \eta \models h$ where $\models$ is the satisfaction relation [21]. Let $\mathcal{A}$ be the set of actions where each action $a \in \mathcal{A}$ is an n-tuple of functions $(a_1, \ldots, a_n)$, updating the current variable evaluation η to the new evaluation $a(\eta)$. Every action is a function $a_i : D_i \to D_i$ which maps the variables evaluation of the current location to the variables evaluation of the next location. The symbol ξ is used to indicate that a variable is not updated; and in vector form $\Xi = \{\xi, \ldots, \xi\}$. If $a_i = \xi$, we say that a_i is a don't care updating of the variable v_i, meaning that it takes it current value, i.e., $a(\eta(v_i)) = \eta(v_i)$.

Definition 1 (Extended Finite Automaton)**.**
An extended finite automaton over a set of variables $\mathcal{V}$ is an 8-tuple

$$E = (L, D, \Sigma, T, L^0, D^0, L^m, D^m),$$

where L is a finite set of discrete locations, $D = D_1 \times \cdots \times D_n$ is the domain of variables, Σ is a nonempty finite set of events (alphabets), $T \subseteq L \times \Sigma \times \mathcal{G} \times \mathcal{A} \times L$ is the *conditional* transition relation, $L^0 \subseteq L$ is the set of initial locations, $D^0 = D_1^0 \times \cdots \times D_n^0$ is the set of variables initial values, $L^m \subseteq L$ is the set of marked (desired) locations, and $D^m \subseteq D$ is the set of marked values of the variables.

The initial variable evaluation, denoted η^0, assigns each variable $v_i \in \mathcal{V}$ to its initial value D_i^0. The notation $\ell \overset{\sigma}{\to}_{g/a} \acute{\ell}$ is used as shorthand for $(\ell, \sigma, g, a, \acute{\ell}) \in T$. If the guard of the transition $\ell \overset{\sigma}{\to}_{g/a} \acute{\ell}$, is a tautology, e.g. $g = \mathbf{T}$ or $g = (v < 1) \vee (v \geqslant 1)$, then we simply write $\ell \overset{\sigma}{\to}_{a} \acute{\ell}$. Similar to DFA, let $\ell \overset{\sigma}{\to}_{g/a}$ denote the existence of a location $\acute{\ell} \in L$ such that $\ell \overset{\sigma}{\to}_{g/a} \acute{\ell}$.

It is assumed that all actions are written as constant functions where the new value of v_i only depends on its previous value. Any transition can be decomposed into multiple transitions of this form. For instance, the transition $\ell \overset{\sigma}{\to}_{x:=y+1} \acute{\ell}$ where $D(y) = \{0, 1\}$ can be decomposed into $\ell \overset{\sigma}{\to}_{y=0/x:=1} \acute{\ell}$ and $\ell \overset{\sigma}{\to}_{y=1/x:=2} \acute{\ell}$.

EFAs can be unfolded to their underlying FAs whose states and transitions are defined as follows.

Definition 2 (FA Semantics of an EFA)**.**
Let $E = (L, D, \Sigma, T, L^0, D^0, L^m, D^m)$ be an EFA over the set of variables $\mathcal{V}$. The FA $G(E)$ is the tuple $(Q_E, \Sigma_E, \mapsto_E, Q_E^0, Q_E^m)$ where $Q_E = L \times D$, $\mapsto_E \subseteq Q \times \Sigma \times Q$ is defined by the following rule:

$$\frac{\ell \overset{\sigma}{\to}_{g/a} \acute{\ell} \wedge \eta \models g}{\langle \ell, \eta \rangle \overset{\sigma}{\mapsto} \langle \acute{\ell}, a(\eta) \rangle},$$

$Q_E^0 = L^0 \times D^0$, and $Q_E^m = L^m \times D^m$.

States of $G(E)$ are the set of reachable states of E and each state consists of a location ℓ together with variable evaluation η. Note that in the definition of transition relation $\mapsto$, if the proposition above the horizontal line holds, then the proposition under the line holds as well (also known as Structured Operational Semantics), namely, whenever the guard g of the conditional transition $\ell \overset{\sigma}{\to}_{g/a} \acute{\ell}$ holds for the variable evaluation η, i.e., $\eta \models g$, then there is a transition in $G(E)$ from state $\langle \ell, \eta \rangle$ to state $\langle \acute{\ell}, a(\eta) \rangle$. Observe that, the

DFA generated directly from a given EFA by constructing the state set as $L \times D$ is not guaranteed to be the canonical recognizer and therefore further reduction needs to be done by using the standard algorithm of minimization [22]. In the sequel, we assume that the DFA obtained by the above transformation is a canonical recognizer of the language represented by the input EFA model.

Since we are interested in deterministic systems, we only focus on deterministic EFAs and, for the sake of brevity, we simply write EFAs for deterministic EFAs.

Definition 3 (Deterministic EFA)**.**
An EFA E is deterministic if $G(E)$ is deterministic, namely, the set of initial states of $G(E)$ is a singleton $\langle \ell^0, \eta^0 \rangle$, where $\ell^0 \in L^0$ and η^0 is initial variable evaluation, and for all transitions $\langle \ell, \eta \rangle \overset{\sigma}{\mapsto} \langle \acute{\ell}, \acute{\eta} \rangle$ and $\langle \ell, \eta \rangle \overset{\sigma}{\mapsto} \langle \grave{\ell}, \grave{\eta} \rangle$ it always implies that $\langle \acute{\ell}, \acute{\eta} \rangle = \langle \grave{\ell}, \grave{\eta} \rangle$.

Given two EFAs E and $\acute{E}$, we say that $\acute{E}$ is a sub-EFA of E, denoted $E_1 \subseteq E_2$, if $\acute{E}$ is obtained from E by removing some locations of E as well as the transitions linked to these locations or removing some transitions of or replacing the guards of some edges of by subguards. Since the transitions in EFAs are conditional, it is not possible to describe the static behavior of the system by following its transitions before evaluating its guards and actions. Therefore, a notion of *execution fragment* that is a series of transitions with guards and actions ending with a location is used to describe the dynamic behavior of EFAs.

Definition 4 (Finite Execution Fragment)**.**
Let $E = (L, D, \Sigma, T, \ell^0, \eta^0, L^m, D^m)$ be an EFA over the set of variables $\mathcal{V}$. A *finite execution fragment* ϱ in E is a series of transitions

$$\varrho = \ell_0 \overset{\sigma_1}{\to}_{g_1/a_1} \ell_1 \overset{\sigma_2}{\to}_{g_2/a_2} \cdots \overset{\sigma_{i+1}}{\to}_{g_{i+1}/a_{i+1}} \ell_{i+1},\ (0 \leqslant i < n),$$

where $n \geqslant 0$. The integer n is the length of the execution fragment.

The $\varrho = \ell_0$ for some $\ell_0 \in L$ is a legal finite execution fragment, henceforth execution fragment, of length $n = 0$. Note that, in execution fragments, we do not explicitly list the selfloops of the empty string ε as they are trivially contained in any EFA. The first and last location of ϱ is denoted by *first*(ϱ) and *last*(ϱ), respectively, *str*(ϱ) denotes $\sigma_1\sigma_2 \ldots \sigma_{i+1}$, and *Loc*($\varrho$) denotes the set of locations $\{\ell_0, \ell_1, \ldots, \ell_{i+1}\}$ $(0 \leqslant i < n)$, that can be reached by following the transitions in ϱ. We call ϱ an initial execution fragment if *first*(ϱ) $\in L^0$ and $\eta_0 = \eta^0$, and a marked execution fragment if *last*(ϱ) $\in L^m$ and $\eta_n \in D^m$. Finally, ϱ is accepted by E if for all transition $\ell_i \overset{\sigma_{i+1}}{\to}_{g_{i+1}/a_{i+1}} \ell_{i+1} \in \varrho$ we have $\eta_i \models g_{i+1}$ and $\eta_{i+1} = a(\eta_i)$.

For two execution fragments ϱ^i, $i = 1, 2$, in E, we say ϱ^1 is a *precedence* of ϱ^2, written $\varrho^1 \sqsubseteq \varrho^2$, if *last*($\varrho^1$) = *first*($\varrho^2$) and $\eta^1 = \eta^2$ where η^1 and η^2 are the current variable evaluations for the locations *last*(ϱ^1) and *first*(ϱ^2), respectively and $\varrho^1 = \varrho^2$ iff *str*(ϱ^1) = *str*(ϱ^2) and for all transitions $(\ell^1_j, \sigma, g^1_{j+1}, a^1_{j+1}, \ell^1_{j+1}) \in \varrho^1$, $g^1_{j+1} \neq \varepsilon$, there exists a transition $(\ell^2_j, \sigma, g^2_{j+1}, a^2_{j+1}, \ell^2_{j+1}) \in \varrho^2$ such that $g^1_{j+1} = g^2_{j+1}$ and $a^1_{j+1} = a^2_{j+1}$ for all $0 \leqslant j < |\varrho^1|$ up to renaming of locations.

EFAs similar to ordinary automata are composed by the extended full synchronous composition (EFSC). By the definition of EFSC, it is assumed that the variables are shared by all EFAs with the same initial variable evaluation.

Definition 5 (EFSC)**.**
Let $E_k = (L_k, D, \Sigma_k, T_k, \ell^0_k, \eta^0, L^m_k, D^m_k),\ k = 1, 2$, be two EFAs over the set of shared variables $\mathcal{V}$. The *Extended Full Synchronous Composition* of E_1 and E_2 is

$$E_1 \| E_2 = (L, D, \Sigma, T, \ell^0, \eta^0, L^m, D^m),$$

where $L = L_1 \times L_2$, $\Sigma = \Sigma_1 \cup \Sigma_2$, T is defined by the rules:

* $\dfrac{\ell_1 \overset{\sigma}{\to}_{1, g_1/a_1} \acute{\ell}_1 \wedge \ell_2 \overset{\sigma}{\to}_{2, g_2/a_2} \acute{\ell}_2 \wedge \sigma \in (\Sigma_1 \cap \Sigma_2)}{\langle \ell_1, \ell_2 \rangle \overset{\sigma}{\to}_{g/a} \langle \acute{\ell}_1, \acute{\ell}_2 \rangle}$

  such that

  (i) $g = g_1 \wedge g_2$,

  (ii) For $i = 1, \ldots, n$:

  $$a_i = \begin{cases} a_{1i} & \text{if } a_{1i} = a_{2i} \\ a_{1i} & \text{if } a_{2i} = \xi \\ a_{2i} & \text{if } a_{1i} = \xi \\ \eta_i & \text{otherwise;} \end{cases}$$

* $\dfrac{\ell_1 \overset{\sigma}{\to}_{1, g_1/a_1} \acute{\ell}_1 \wedge \acute{\ell}_2 = \ell_2 \wedge \sigma \in (\Sigma_1 - \Sigma_2)}{\langle \ell_1, \ell_2 \rangle \overset{\sigma}{\to}_{g_1/a_1} \langle \acute{\ell}_1, \acute{\ell}_2 \rangle}$

* $\dfrac{\ell_2 \overset{\sigma}{\to}_{2, g_2/a_2} \acute{\ell}_2 \wedge \acute{\ell}_1 = \ell_1 \wedge \sigma \in (\Sigma_2 - \Sigma_1)}{\langle \ell_1, \ell_2 \rangle \overset{\sigma}{\to}_{g_2/a_2} \langle \acute{\ell}_1, \acute{\ell}_2 \rangle}$,

$L^0 = L^0_1 \times L^0_2$, and $L^m = L^m_1 \times L^m_2$.

Note that, if the action functions of E_1 and E_2 try to update a shared variable to different values, the variable is, by default, not updated. A situation where two values are conflicting, is usually a consequence of bad modeling. In this work, in order to avoid conflicting variables, we assume that for any two conditional transitions in the system with the same label, say $\ell_1 \overset{\sigma}{\to}_{g/a} \ell_2$ and $\acute{\ell}_1 \overset{\sigma}{\to}_{\acute{g}/\acute{a}} \acute{\ell}_2$, if both $a, \acute{a} \neq \Xi$ always implies $a(\eta) = \acute{a}(\acute{\eta})$. In general, this assumption may restrict the modeling using EFA. But in practice, shared events in ordinary DFAs, which are used for mutual exclusion, precedence relations and supervisor implementation, are replaced by guards on shared variables in EFAs.

C. Supervisory Control of EFA

SCT is a formal framework for the modeling and control of DES consisting of a plant and a specification. In this work, we use the symbolic algorithm presented in [8] to efficiently synthesize a nonblocking supervisor. The algorithm iteratively strengthens the guards on conditional transitions to avoid forbidden or blocking states. Given a DES control problem, we assume that all events are controllable, the plant is modeled by an EFA P, and the specification by an EFA Sp. The specification can be represented, without loss of generality, by a set of forbidden locations which can be obtained by a refined plant model R with the same behavior as P such that the executions not allowed in Sp end up in

certain forbidden locations in R. See [8] for more elaboration on refinement.

From now on, we assume that the plant model is given as the refined EFA R and the specification is given as the set of forbidden locations $L_f \subset L_R$. Let us denote the set of safe locations by $L_s = L - L_f$ and recall the set of reachable states Q_R in $G(R)$. A state $q = \langle \ell, \eta \rangle \in Q_R$ is a forbidden state iff $\ell \in L_f$, otherwise, q is a safe state. In the sequel, R_s denotes the EFA obtained from R by assigning $\mathbf{F}$ to the guard g of every transition $\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}$ for which $\acute{\ell} \in L_f$, i.e., $\acute{\ell}$ is a forbidden location. R_s is constructed such that $R_s \subseteq R$ and is called the safe sub-EFA of R.

Definition 6 (Nonblocking, Safety)**.**
[8] Let R be an EFA, L_f its set of forbidden locations, and R_s its safe sub-EFA. A reachable state $q \in Q_R$ is: (a) *nonblocking* if there exists a state $p \in Q_R^m$ such that $q \overset{s}{\mapsto} p$ for some string $s \in \Sigma^*$; (b) safe if $q \in Q_{R_s}$. The EFA R is, respectively, nonblocking and safe if every reachable state of R is, respectively, nonblocking and safe.

A supervisor S for R can be seen as a function $S : T \to \mathcal{G}$ which maps each transition to a supervision guard such that $S(\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}) \preceq g$ if $\sigma \in \Sigma_c$, and $S(\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}) = g$ if $\sigma \in \Sigma_u$. Let R^S denote the sub-EFA obtained from R by replacing its guards by those provided by S. Then, S is said to be nonblocking if R^S is nonblocking and safe if R^S is safe. In case R^S is blocking, a search will be performed to find a safe and nonblocking supervisor S such that $R^S \subseteq R_s$. Let $S(R, L_f)$ denote the set of nonblocking and safe supervisor candidates of R, then $S^\uparrow := supS(R, L_f)$, is the *most permissive nonblocking and safe supervisor* compared to any other supervisor in $S(R, L_f)$ when the latter is nonempty. The $R^{S^\uparrow}$ is called the supremal nonblocking sub-EFA of R_s.

$R^{S^\uparrow}$ is calculated by the Supervisory Synthesis for EFA (SSEFA) [8] using a fixed-point iteration method. Given a refined EFA R and a set $L_f \subset L$ of forbidden location, SSEFA(R, L_f) computes stronger, maximally permissive, guards for the transitions of R in N steps such that the obtained EFA is nonblocking.

III. EFA Projection

Traditionally, brute-force computation is used for verification and coordination [20]. This we wish to avoid since the nonblocking supervisory control problem in SCT is NP-hard.

Abstraction introduces hierarchy into the system structure, as it reports only the events shared with other subsystems and conceals the rest. The fewer the reported events, the greater state reduction will be achieved. Natural projection [23] with observer property is a language-theoretic operation, which cannot be used for EFAs with conditional transitions before evaluating the guards and the actions. In order to use model abstraction using projection, we substitute the natural projection with transition projection to be able to abstract the system by their transition systems. In this section a DES is assumed to consist of a group of simple plant EFA components, subject to a conjunction of modular control specifications. Before introducing the transition projection we need the following notations.

For an event σ, let $Act(\sigma) \subseteq \mathcal{A}$ and $Con(\sigma) \subseteq \mathcal{G}$ be the sets of actions and guards, respectively, retrieved from all transitions labeled with σ. Note that, by the assumption after Definition 5, the set $Act(\sigma)$ is a singleton a_σ.

Definition 7 (Local Event)**.**
For an EFA $E_i, i \in \mathbf{n}$, over the set of shared variables $\mathcal{V}$, an event $\sigma \in \Sigma_i$ is *local* to E_i if for all $j \in \mathbf{n}$ we have (i) $\sigma \in \Sigma_i - \bigcup \Sigma_j$ $(j \neq i)$, (ii) $(\forall g \in Con(\sigma))$ g is tautology, (iii) $(\forall g \in \mathcal{G}_j)$ $\eta \models g \Leftrightarrow a_\sigma(\eta) \models g$.

Here, condition (i) guarantees that the event σ only pertains to E_i and not to other EFAs E_j $(j \neq i)$, (ii) ensures that guards on any transition labeled by σ is a tautology when clear from context; hence σ can cause the transition to occur at any time, and (iii) guarantees that the execution of action a_σ has no effect on any guards evaluation. Any transition labeled with a local event is called a *local transition*, and similarly any execution fragment is local if its transitions are all local.

For an EFA E over the set of variables $\mathcal{V}$ and the set of events Σ, the transition projection $\bar{P}$ for the conditional transition relation T and the set $\Sigma_\ell \subseteq \Sigma$ is defined as follows:

$$\bar{P} : T \times \Sigma_\ell \to T$$

where for every transition $\ell \xrightarrow{\sigma}_{g/a} \acute{\ell} \in T$ and $\gamma \in \Sigma_\ell$

$$\bar{P}[\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}, \varepsilon] = \ell \xrightarrow{\sigma}_{g/a} \acute{\ell}$$

$$\bar{P}[\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}, \gamma] = \begin{cases} \ell \xrightarrow{\sigma}_{g/a} \acute{\ell}, & \sigma \neq \gamma \\ \ell \xrightarrow{\varepsilon}_{g/a} \acute{\ell}, & \sigma = \gamma \end{cases}$$

The transition projection $\bar{P}$ replaces the label of transitions labeled by events in Σ_ℓ with symbol ε. In effect, an EFA is allowed to make a transition spontaneously, without receiving an input event. Extending T to its power set $Pwr(T)$, we get $\bar{P} : Pwr(T) \times \Sigma_\ell \to Pwr(T)$ such that for any $\tau \in \Sigma_\ell$, $N \subseteq T$: $\bar{P}(N, \tau) = \{\bar{P}(\ell \xrightarrow{\sigma}_{g/a} \acute{\ell}, \tau) | \ell \xrightarrow{\sigma}_{g/a} \acute{\ell} \in N\}$. If we further extend Σ_ℓ to its power set $Pwr(\Sigma_\ell)$, $\bar{P}$ becomes $\bar{P} : Pwr(T) \times Pwr(\Sigma_\ell) \to Pwr(T)$ such that for $A \in \Sigma_\ell$, $N \subseteq T$: $\bar{P}(N, A) = \bigcup\{\bar{P}(N, \tau) | \tau \in A\}$. If the effect of $\bar{P}$ on T is understood then $\bar{P}[T, \Sigma_\ell]$ may be written $\bar{P}_{\Sigma_\ell} T$ and if $\bar{P}$ is defined then $\bar{P}T$. A projected (observer) EFA $\tilde{E}$ of an EFA E whose transitions are projected by the transition projection $\bar{P} : T \times \Sigma_\ell \to T$ can be constructed as follows. Let $\mathcal{S}_\varepsilon(\ell)$ be the set of ε-closure of a location ℓ in E. $\mathcal{S}_\varepsilon(\ell)$ is constructed recursively by finding every location that can be reached from ℓ along any path whose transitions are all labeled ε. Formally, (1) $\ell \in \mathcal{S}_\varepsilon(\ell)$, (2) $(\forall \acute{\ell} \in \mathcal{S}_\varepsilon(\ell))$ $\acute{\ell} \xrightarrow{\varepsilon}_{g/a} \grave{\ell} \Rightarrow \grave{\ell} \in \mathcal{S}_\varepsilon(\ell)$. The location set of $\tilde{E}$ will be denoted by $\tilde{L}$, with element $\tilde{\ell}$ that label ε-closure subsets of E. Evidently ε-closure subsets might be nondeterministic, i.e., there is more than one outgoing transition with the same event label in $\Sigma - \Sigma_\ell$ from these subsets. A deterministic model can then be achieved as follows [19]. Define the initial location subset $\tilde{\ell}^0 := \mathcal{S}_\varepsilon(\ell^0)$. Choose $\sigma_1 \in \Sigma - \Sigma_\ell$ and define $\tilde{\ell}^1 := \bigcup_{\ell \in \tilde{\ell}^0} \{\mathcal{S}_\varepsilon(\acute{\ell}) \mid (\ell, \sigma_1, g, a_\sigma, \acute{\ell}) \in T\}$. Define $\tilde{\ell}^2$ similarly, from $\tilde{\ell}^0$ and $\sigma_2 \in \Sigma - \Sigma_\ell - \{\sigma_1\}$, and repeat until $\Sigma - \Sigma_\ell$ is exhausted. The subset obtained at any step is

discarded if it is empty or if it appeared previously. This process yields a list of distinct nonempty subsets $\tilde{\ell}^0, \tilde{\ell}^1, \ldots, \tilde{\ell}^k$ and one-step 'subset' transitions of form $(\tilde{\ell}^0, \sigma, g_\sigma, a_\sigma, \tilde{\ell}^i)$, $\sigma \in \Sigma - \Sigma_\ell$, $i \in \{0, 1, \ldots, k\}$, augmented with the action a_σ (recall that by the assumption, all transitions with the same event label has the the same action) and the guard g_σ that is the disjunction of guards on all transitions from the locations in subset $\tilde{\ell}^0$ to the locations in subset $\tilde{\ell}^i$, more specifically, g_σ is defined for all $\ell \in \tilde{\ell}^0$ and for all $\acute{\ell} \in \tilde{\ell}^i$ such that $(\ell, \sigma, g, a_\sigma, \acute{\ell}) \in T$, and $g_\sigma = g_\sigma \vee g$ (initially assigned to **F**). The procedure is repeated recursively for each of the subsets $\tilde{\ell}^1, \tilde{\ell}^2, \ldots$ and each $\sigma \in \Sigma - \Sigma_\ell$, until no new subset transitions are obtained. The result is the projected EFA

$$\tilde{E} = (\tilde{L}, D, \tilde{\Sigma}, \tilde{T}, \tilde{\ell}^0, \eta^0, \tilde{L}^m, D^m),$$

where $\tilde{L}$ is the final list $\{\tilde{\ell}^0, \tilde{\ell}^1, \ldots, \}$, $\tilde{L}^m := \{\tilde{\ell} \in \tilde{L} \mid \tilde{\ell} \cap L^m \neq \emptyset\}$, and $(\tilde{\ell}, \sigma, g_\sigma, a_\sigma, \tilde{\ell}') \in \tilde{T}$ iff $(\ell, \sigma, g, a_\sigma, \acute{\ell}) \in T$ for some $\ell \in \tilde{\ell}, \acute{\ell} \in \tilde{\ell}', \sigma \in \Sigma - \Sigma_\ell$. Let the projected EFA, by the set of local events as described above, be achieved by the function $\hat{P} : E \times \Sigma_\ell \to E$, that is, $\hat{P}[E, \Sigma_\ell] = \tilde{E}$.

Turning to the supervisory control problem, consider a system consisting of two EFA components, E_1 and E_2, over event sets Σ_i $(i = 1, 2)$. To obtain a reduction of the system, we could first compute the systems global behavior $E_1 \| E_2$ and then its transition projection. When, however, the local events of the two components are all defined the result is obtained more economically from reductions of the components, according to the following proposition. This result is central to our method.

Proposition 1.
Let $E_k = (L_k, D, \Sigma_k, T_k, \ell_k^0, \eta^0, L_k^m, D_k^m), k = 1, 2$, be two EFAs over the set of shared variables $\mathcal{V}$. Consider T as the set of transition relations for $E_1 \| E_2$ and let $\Sigma_\ell \subseteq \Sigma := \Sigma_1 \cup \Sigma_2$. Define $\bar{P} : T \times \Sigma_\ell \to T$ and $\bar{Q}_i : T_i \times (\Sigma_i \cap \Sigma_\ell) \to T_i$ $(i = 1, 2)$. If Σ_ℓ is the set of local events, then the EFA projection

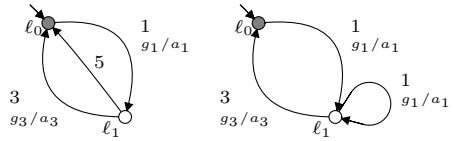
$$\hat{P}[E_1 \| E_2, \Sigma_\ell] = \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell].$$

Proof: See [24]. ∎

The extension to an arbitrary number of synchronized factors is straightforward and is left out.

IV. EFA Observer

Consider a DES described by EFA E. Given a set of local events, we can define the transition projection $\bar{P} : T \times \Sigma_\ell \to T$ and then the EFA projection $\hat{P}[E, \Sigma_\ell]$. Crucial to successful model abstraction using transition projection is that the projected system contains necessary and sufficient information needed for reliable representation of the nonblocking property. In other words, the EFA projection $\hat{P}$ may remove critical information and be inconsistency with the original DES with respect to nonblocking. For instance, the projection of a blocking DES could be nonblocking, so a nonblocking supervisor designed from the EFA projection could result in a blocking supervisor for the original DES. To avoid this pitfall, one must carefully select the local events of a DES.



(a) E (b) $\tilde{E}$

Fig. 1: $\tilde{E}$ is the projection of E but not an E-observer.

A "good" selection of local events is whenever a projected EFA reaches a location by $\bar{P}\varrho_s$ and then a marked location by $\bar{P}\varrho_t$, the original system, must be able to reach a marked location from $\varrho_{\acute{s}}$, via some $\varrho_{\acute{t}}$ such that $\bar{P}\varrho_{\acute{s}} = \bar{P}\varrho_s$ and $\bar{P}\varrho_{\acute{t}} = \bar{P}\varrho_t$.

Definition 8 (E-observer)**.**
Assume a nonblocking EFA E and let $\Sigma_\ell \subseteq \Sigma$ be the subset of events. The transition projection $\bar{P} : T \times \Sigma_\ell \to T$ is an *E-observer*, if for all initial execution fragments ϱ_s and $\varrho_{\acute{s}}$ and for all marked execution fragment ϱ_t in E such that $\varrho_s \sqsubseteq \varrho_t$ and $\bar{P}\varrho_{\acute{s}} = \bar{P}\varrho_s$, there exists a marked execution fragment $\varrho_{\acute{t}}$ in E such that $\varrho_{\acute{s}} \sqsubseteq \varrho_{\acute{t}}$ and $\bar{P}\varrho_{\acute{t}} = \bar{P}\varrho_t$.

Example 1. *Consider EFAs E and $\tilde{E}$ in Fig. 1 and assume the set of local events $\Sigma_\ell = \{5\}$. The shaded circle is the marked location. Define the transition projection $\bar{P} : T \times \Sigma_\ell \to T$ and let $\varrho_s = \ell_0 \xrightarrow{1}_{g_1/a_1} \ell_1$, $\varrho_t = \ell_1 \xrightarrow{3}_{g_3/a_3} \ell_0$, and $\varrho_{\acute{s}} = \ell_0 \xrightarrow{1}_{g_1/a_1} \ell_1 \xrightarrow{5} \ell_0$ such that $\varrho_s \sqsubseteq \varrho_t$ and $\bar{P}\varrho_s = \bar{P}\varrho_{\acute{s}}$. We cannot further find any execution fragment, say $\varrho_{\acute{t}}$ in E such that $\varrho_{\acute{s}} \sqsubseteq \varrho_{\acute{t}}$ and $\bar{P}\varrho_t = \bar{P}\varrho_{\acute{t}}$. Thus $\bar{P}$ is not an E-observer.*

Let denote $E = \emptyset$ when there is no outgoing transition from the initial location of E.

Lemma 1.
In the notation of Proposition 1, define the transition projection $\bar{P}_i : T \times (\Sigma_j - \Sigma_i) \to T$ for $i, j = 1, 2$ and $j \neq i$. For the EFAs E_1 and E_2 if $\Sigma - \Sigma_\ell \neq \emptyset$, $(\exists \ell_i \xrightarrow{\sigma}_{g/a} \acute{\ell}_i \in T_i)$ $\sigma \in \Sigma - \Sigma_\ell$ for some $\ell_i, \acute{\ell}_i \in L_i$ $(i = 1, 2)$ we have $E_1 \| E_2 \neq \emptyset \Leftrightarrow \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell] \neq \emptyset$.

Proof: ($\Rightarrow$) We know $E_1 \| E_2 \neq \emptyset$. Also, there exists $\varrho \in E_1 \| E_2$ with some transitions labeled by events in $\Sigma - \Sigma_\ell$. Applying $\bar{P}$ we get $\bar{P}\varrho \in \hat{P}[E_1 \| E_2, \Sigma_\ell]$. By the hypothesis assumption $\Sigma - \Sigma_\ell \neq \emptyset$ which implies that $\hat{P}[E_1 \| E_2, \Sigma_\ell] \neq \emptyset$ and by Proposition 1, $\hat{P}[E_1 \| E_2, \Sigma_\ell] = \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell] \neq \emptyset$.
($\Leftarrow$) We know $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell] \neq \emptyset$ and by Proposition 1 we also know that $\hat{P}[E_1 \| E_2, \Sigma_\ell] \neq \emptyset$. Then, taking any execution fragment $\acute{\varrho} \in \hat{P}[E_1 \| E_2, \Sigma_\ell]$ there must be an execution fragments $\varrho \in E_1 \| E_2$ such that $\bar{P}\varrho = \acute{\varrho}$ and therefore, $E_1 \| E_2 \neq \emptyset$. ∎

In the sequel, we assume that $\Sigma_\ell \subset \Sigma := \cup \Sigma_i$, $(\exists \ell_i \xrightarrow{\sigma}_{g/a} \acute{\ell}_i \in T_i)$ $\sigma \in \Sigma - \Sigma_\ell$ $(i \in \mathbf{n})$. Note that if Σ_ℓ is equal to Σ or $\emptyset$, $\bar{P}$ is automatically an E-observer.

In a system consisting of more than one plant component, it would be more economical to check the E-observer property component-wise without computing the synchronous product first. Proposition 2 presents a sufficient condition for

this simplification to be valid.

Proposition 2.
Let $E_k = (L_k, D, \Sigma_k, T_k, \ell_k^0, \eta^0, L_k^m, D_k^m), k = 1,2$ *be two nonblocking EFAs over the set of shared variables* $\mathcal{V}$. *Consider* T *as the set of transition relations for* $E_1 \| E_2$. *Define the transition projections* $\bar{P} : T \times \Sigma_\ell \to T$ *and* $\bar{Q}_i : T_i \times (\Sigma_i \cap \Sigma_\ell) \to T_i$ $(i = 1,2)$ *where* $\Sigma_\ell \subset \Sigma := \Sigma_1 \cup \Sigma_2$. *If* Σ_ℓ *is the set of local events and for both* $i = 1,2, \bar{Q}_i$ *is an* E_i*-observer, then* $\bar{P}$ *is an* $E_1 \| E_2$*-observer.*

Proof: See [24]. ■

As we establish a "reliable interface" for EFAs by introducing E-observer, the interaction between two complex systems may be examined through their projections rather than their global behavior. Since the EFA models of $\hat{P}[E_i, \Sigma_i \cap \Sigma_\ell]$ are smaller than those of E_i, we may save significant computational effort, in accordance with the following.

Theorem 1 (Synchronously Nonconflicting Criterion)**.**
Let $E_k = (L_k, D, \Sigma_k, T_k, \ell_k^0, \eta^0, L_k^m, D_k^m), k = 1,2$, *be two EFAs with the set of shared variables* $\mathcal{V}$ *and let* $\Sigma_\ell \subset \Sigma := \Sigma_1 \cup \Sigma_2$ *be the set of local events. If* $\bar{Q}_i : T_i \times (\Sigma_i \cap \Sigma_\ell) \to T_i$ *are* E_i*-observer* $(i = 1,2)$, *then* $E_1 \| E_2$ *is nonblocking if and only if* $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$ *is nonblocking.*

Let $\bar{P}_i : T \times (\Sigma_j - \Sigma_i) \to T$ $(j \neq i)$, $\bar{Z} : T \times ((\Sigma_1 \cup \Sigma_2) - (\Sigma_1 \cap \Sigma_2)) \to T$, and $\bar{R}_i := \bar{Q}_i \circ \bar{P}_i (i, j = 1,2)$.
(If) Let ϱ_s be an initial execution fragment in $E_1 \| E_2$. We must show that there exists a marked execution fragment ϱ_t such that $\varrho_s \sqsubseteq \varrho_t$. Apply $\bar{P}_i$ to ϱ_s, we get $\bar{P}_i \varrho_s \in E_i$ $(i = 1,2)$. We also know that $\bar{P}\varrho_s \in \hat{P}[E_1 \| E_2, \Sigma_\ell]$. Because of the assumption that Σ_ℓ is the set of local events and by Proposition 1, $\bar{P}\varrho_s \in \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$. Then, by Proposition 2 there must exist a marked execution fragment $\acute{\varrho}_t \in \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$ such that $\bar{P}\varrho_s \sqsubseteq \acute{\varrho}_t$. Applying $\bar{R}_i$ on both sides, we get $\bar{R}_i \bar{P} \varrho_s$ and $\bar{R}_i \acute{\varrho}_t$.We have $\bar{R}_i \circ \bar{P} = \bar{Q}_i \circ \bar{P}_i$ $(i = 1,2)$. Consequently, both $\bar{Q}_i \bar{P}_i \varrho_s$ and $\bar{R}_i \acute{\varrho}_t$ are in $\hat{Q}_i[E_i, \Sigma_i \cap \Sigma_\ell] (i = 1,2)$. Since $\bar{P}_i \varrho_s \in E_i$ and $\bar{Q}_i$ is an E_i-observer, there exists a marked execution fragment $\varrho_{w_i} \in E_i$ such that $\bar{P}_i \varrho_s \sqsubseteq \varrho_{w_i}$ and $\bar{Q}_i \varrho_{w_i} = \bar{R}_i \acute{\varrho}_t$. Applying $\bar{P}_j (j = 1,2; j \neq i)$ to both sides of this equation, we get $\bar{P}_j \bar{Q}_i \varrho_{w_i} = \bar{P}_j \bar{R}_i \acute{\varrho}_t = \bar{Z} \acute{\varrho}_t$ and $\bar{P}_j \circ \bar{Q}_i = \bar{P}_j$. This implies that $\bar{P}_2 \varrho_{w_1} = \bar{Z} \acute{\varrho}_t = \bar{P}_1 \varrho_{w_2}$. Constructing the set $\Pi := \{\varrho_w \in E_1 \| E_2 \mid \bar{P}_1 \varrho_w = \varrho_{w_1} \wedge \bar{P}_2 \varrho_w = \varrho_{w_2}\}$ by Lemma 1 we know that $\Pi \neq \emptyset$. Taking any execution fragment from this set, say $\varrho_w \in \Pi$ where $\varrho_s \sqsubseteq \varrho_w$ we observer that ϱ_w is marked as required.
(Only if) According to the assumption $E_1 \| E_2$ is nonblocking and therefore, for any initial execution fragment ϱ_s there exists a marked execution fragment ϱ_t such that $\varrho_s \sqsubseteq \varrho_t$. Recall the Proposition 1, apply $\bar{P}$ on both ϱ_s and ϱ_t, we get, respectively, $\bar{P}\varrho_s$ and $\bar{P}\varrho_t$ in $\hat{P}[E_1 \| E_2, \Sigma_\ell] = \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$. Since $\bar{P}$ is $E_1 \| E_2$-observer, there must exist a marked execution fragment $\acute{\varrho}_t \in \hat{P}[E_1 \| E_2, \Sigma_\ell]$ such that $\bar{P}\varrho_s \sqsubseteq \acute{\varrho}_t$ and $\bar{P}\acute{\varrho}_t = \bar{P}\varrho_t$. Therefore, for any execution fragment $\bar{P}\varrho_s \in \hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$ there exists a marked execution fragment $\acute{\varrho}_t$ such that $\bar{P}\varrho_s \sqsubseteq \acute{\varrho}_t$ and therefore, $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$ is also nonblocking. ■

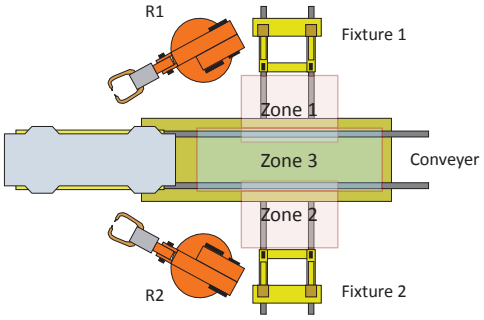


Fig. 2: The robot workcell.

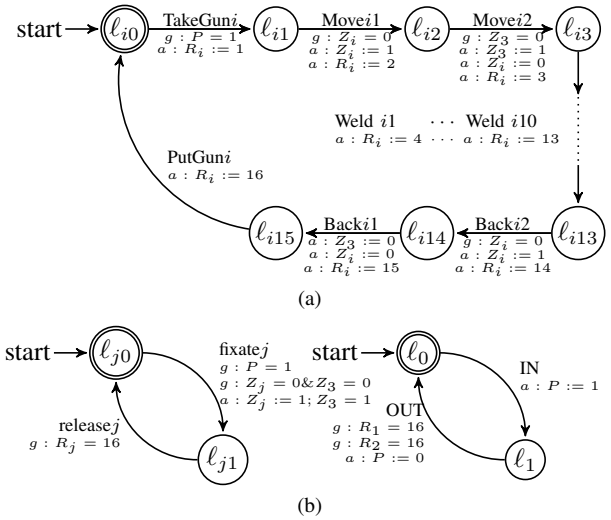


Fig. 3: EFA models of (a) Robot i for $i = 1,2$ and (b) on the left is the fixture j for $j = 1,2$ and on the right is the conveyor.

In case two EFAs E_1 and E_2 are synchronously conflicting, a third EFA E must be introduced to resolve the conflict. Instead of computing the EFA E directly from the two EFAs themselves, we can perform this computation through their abstractions.

Proposition 3.
In the notation of Proposition 2, if there exists an EFA E *such that* $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell] \| \hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell] \| E$ *is nonblocking then* $E_1 \| E_2 \| E$ *is also nonblocking.*

Proof: See [24]. ■

As long as E can resolve the conflict between $\hat{Q}_1[E_1, \Sigma_1 \cap \Sigma_\ell]$ and $\hat{Q}_2[E_2, \Sigma_2 \cap \Sigma_\ell]$, it can resolve the conflict between E_1 and E_2.

V. Example

A. Robot Workcell

The proposed approach is applied to the nonblocking supervisory control of a simplified mid-sized robot workcell. The robot workcell consists of two robots, two fixtures and a conveyor in the configuration shown in Fig. 2. The system operates as follows: car body is supplied by the conveyor; Fixture 1 and 2 fixate two plates on each side of the car body; each robot takes a welding gun and welds

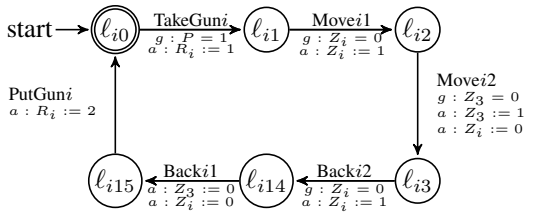


Fig. 4: Projected EFA models of Robot i for $i = 1, 2$.

TABLE I: Optimal nonblocking supervisory synthesis results of the manufacturing workcell example

	Reachable States	Supervisor States
Original Models	211	163
Abstracted Models	111	63

10 geometry points; the fixtures then release the plates; and the conveyor transfers the car body to the next station. The specifications for zones are that Zone 1,2, and 3 can only be occupied by one device at a time. In order to model the zones, variables Z_1, Z_2, and Z_3 with the domain of $D(Z_i) = \{0, 1\}$ for $i = 1, 2, 3$ are introduced. Moreover, to model the workcell flow, variables P with the domain of $\{0, 1\}$ and $R_j (j = 1, 2)$ with the domain of $\{0, 1, \ldots, 16\}$ are used to indicate the availability of car body and the state of robots, respectively. Fig. 3(a) illustrates the EFA models of the robots and Fig. 3(b) shows the EFA models of (left) fixtures and (right) the conveyor. First, the local events in the system is found by checking the observer conditions for each plant. The events which satisfied the condition were then $\Sigma_\ell = \{$*Weld 1, ..., Weld 10*$\}$. Observe that each event in the set Σ_ℓ is (i) unique to their EFA plant, (ii) has a guard that is true, and (iii) has no action that evaluates any guard in the system to true. Therefore, are used to abstract the plant models. The projected plant EFAs of Robot 1 and 2 are shown in Fig. 4. Using the abstracted EFAs, the supervisor can now with less memory and computational effort be obtained by the DES tool Supremica. Table I shows the result of nonblocking supervisory synthesis for both original and abstracted models.

VI. Conclusion

In this paper we have extended previous work on model abstraction by natural projection with a modified observer property to include the EFA modeling formalism. Transition projection is introduced to substitute natural projection for EFAs by projecting the conditional transitions without knowing its underlying language. We independently compute the projection of the low-level components without regarding their mutual conflict. Subsequently, to reduce computational complexity, we compute the high-level coordinators based only on abstracted models of the low-level components. Effective and consistent model abstraction is accomplished through transition projections with the observer property. A robot workcell example demonstrates the computational effectiveness and practical usage of the proposed approach. A special case of this abstraction, including additional structural reduction, has been applied on a large-scale manufacturing workcell [18], where more than 98% of the computational time and space has been saved.

References

[1] P. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proceedings of IEEE, Special Issue on Discrete Event Dynamic Systems*, vol. 77, no. 1, pp. 81–98, 1989.

[2] M. Fabian and A. Hellgren, "PLC-based Implementation of Supervisory Control for Discrete Event Systems," in *37th Decision and Control*, Tampa, FL, USA, 1998.

[3] X.-R. Cao, G. Cohen, A. Giua, W. M. Wonham, and J. H. van Schuppen, "Unity in Diversity, Diversity in Unity: Retrospective and Prospective Views on Control of Discrete Event Systems," *Discrete Event Dynamic Systems*, vol. 12, pp. 253–264, 2002.

[4] M. Skoldstam, K. Åkesson, and M. Fabian, "Modeling of discrete event systems using finite automata with variables," *2007 46th IEEE Conference on Decision and Control*, pp. 3387–3392, 2007.

[5] S. Miremadi, K. Åkesson, and B. Lennartson, "Supervisor Computation and Representation: A Case Study," 2010.

[6] B. Lennartson, K. Bengtsson, C. Yuan, K. Andersson, M. Fabian, P. Falkman, and K. Åkesson, "Sequence planning for integrated product, process and automation design," *Automation Science and Engineering, IEEE Tran.*, vol. 7, no. 4, pp. 791–802, Oct. 2010.

[7] S. Miremadi, B. Lennartson, and K. Åkesson, "A BDD-Based Approach for Modeling Plant and Supervisor by Extended Finite Automata," *IEEE Transactions on Control Systems Technology*, 2011.

[8] L. Ouedraogo, R. Kumar, R. Malik, and K. Åkesson, "Nonblocking and Safe Control of Discrete-Event Systems Modeled as Extended Finite Automata," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 3, pp. 560–569, Jul. 2011.

[9] K. Åkesson, M. Fabian, H. Flordal, and R. Malik, "Supremica—an integrated environment for verification, synthesis and simulation of discrete event systems," in *Proceedings of WODES'08*, Ann Arbor, MI, USA, 2006, pp. 384–385.

[10] P. Gohari and W. M. Wonham, "On the complexity of supervisory control design in the RW framework." *IEEE transactions on systems, man, and cybernetics.*, vol. 30, no. 5, pp. 643–52, Jan. 2000.

[11] M. H. Queiroz, J. E. R. Cury, and M. de Queiroz, "Modular control of composed systems," in *American Control Conference*, vol. 6, no. June. American Autom. Control Council, Jun. 2000, pp. 4051–4055.

[12] K. Schmidt, T. Moor, and S. Perk, "A Hierarchical Architecture for Nonblocking Control of Discrete Event Systems," in *Proceedings of the IEEE Conference on Control and Automation Intelligent Control.* IEEE, 2005, pp. 902–907.

[13] R. Leduc, B. Brandin, M. Lawford, and W. M. Wonham, "Hierarchical interface-based supervisory Control-part I: serial case," in *IEEE Transactions on Automatic Control*, vol. 50, no. 9, Orlando, FL, USA, Sep. 2005, pp. 1322–1335.

[14] R. Leduc, M. Lawford, and W. M. Wonham, "Hierarchical interface-based supervisory control-part II: parallel case," in *IEEE Transactions on Automatic Control*, vol. 50, no. 9, Sep. 2005, pp. 1336–1348.

[15] K. C. Wong and W. M. Wonham, "On the Computation of Observers in Discrete-Event Systems," *Discrete Event Dynamic Systems*, vol. 14, no. 1, pp. 55–107, Jan. 2004.

[16] L. Feng and W. M. Wonham, "Supervisory control architecture for discrete-event systems," *Automatic Control, IEEE Transactions on*, vol. 53, no. 6, pp. 1449–1461, 2008.

[17] ——, "On the Computation of Natural Observers in Discrete-Event Systems," *Discrete Event Dynamic Systems*, vol. 20, no. 1, pp. 63–102, Oct. 2008.

[18] M. R. Shoaei, S. Miremadi, K. Bengtsson, and B. Lennartson, "Reduced-order synthesis of operation sequences," in *ETFA2011*. IEEE, Sep. 2011, pp. 1–8.

[19] W. M. Wonham, *Supervisory Control of Discrete Event Systems*, Toronto, Canada, 2011.

[20] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. Springer, 2008.

[21] C. Baier and J.-P. Katoen, *Principles of Model Checking*. The MIT Press, 2008.

[22] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd ed., ser. Series in Computer Science. Addison-Wesley, 2001.

[23] L. Feng, "Computationally efficient supervisor design for discrete-event systems," Doctor of Philosophy, University of Toronto, 2007.

[24] M. R. Shoaei, L. Feng, and B. Lennartson, "Supervisory Control of Extended Finite Automata using Transition Projection," Chalmers University of Technology, Tech. Rep., 2012. [Online]. Available: http://publications.lib.chalmers.se/cpl/record/index.xsql?pubid=155706