Supervisory Control of Asynchronous and Hierarchical Finite State Machines

Benoit Gaudin, Hervé Marchand

MODULAR SUPERVISORY CONTROL OF ASYNCHRONOUS AND HIERARCHICAL FINITE STATE MACHINES

B. Gaudin, H. Marchand

VerTeCs Team, Irisa, Campus Universitaire de Beaulieu, 35042 Rennes, France
E-mail: {bgaudin,hmarchan}@irisa.fr, Fax: +33 2 99 84 71 71

Keywords: Supervisory Control Theory, structured FSM, structured Supervisor, modularity and non-blocking.

Abstract

In this paper, modular supervisory control of a class of Discrete Event Systems is investigated. Discrete event systems are modeled by a Hierarchical Finite State Machine. The basic problem of interest is to solve the State Avoidance Control Problem. We provide algorithms that, based on a particular decomposition of the set of forbidden configurations, locally solve the control problem (i.e. on each component without computing the whole system) and produce a global supervisor ensuring the desired property. This kind of objectives may be useful to perform dynamic interactions between different parts of a system.

1 Introduction

In this paper, we are interested in the Supervisory Control Problem [9] for structured Discrete Event Systems (DES). The system to be controlled is modeled as a collection of Finite State Machines (FSM) that behave asynchronously or by Hierarchical Finite State Machines (by adding nested FSMs). In many applications and control problems, FSMs are the starting point to model fragments of a large scale system, which usually consists of the composition and of the nesting of many different sub-systems. Further, the standard way of applying the supervisor synthesis methodology to such systems is by expanding them to ordinary state machines and by using classical synthesis tools on the resulting FSM. However, knowing that the synthesis algorithms are polynomial in the number of states of the systems and that the number of states of the global systems grows exponentially with the number of parallel and nested sub-systems, it seems important to design algorithms that perform the controller synthesis phase by taking advantage of its structure without expanding the system.

Several approaches have been considered in the literature. In [11], the method consists in dividing the global control objectives into sub-objectives and to perform the controller synthesis phase w.r.t. these sub-objectives. In [4], the authors take advantage of the structure of the system in order to decompose the objectives according to the sub-systems, that are assumed to behave asynchronously. Further, it is sufficient to compute supervisors separately on each sub-system. However, when the control objective is used to express (forbidden) interactions between asynchronous sub-systems, the method proposed in [4] cannot be efficiently used. In order to take into account nested behaviors, some techniques based on model aggregation methods [10, 2] have been proposed to deal with hierarchical control problems. In this paper, we are interested in applying existing techniques to a structured system. We consider here a multi-level hierarchy model in the spirit of [1], who introduced Hierarchical State Machines which constitute a simplified version of STATECHARTS (see also [5, 7, 6] for other works dealing with control and hierarchy. However, in these methods, even if some computations are made locally, the structure of the supervisor does not reflect the one of the plant).

The remainder of this paper is organized as follows. Section 2 consists of a presentation of the basic model on which control will be applied as well as a review of the classical controller synthesis methodology [9]. In section 3, we propose a modular methodology to solve the State Avoidance Control Problem for a plant modeled as a collection of asynchronous sub-plants. The supervisor described in Section 3.2 could be blocking. Then, in section 3.4, we provide sufficient conditions under which the obtained controlled system is non-blocking. Finally, in Section 4, after a brief presentation of the HFSM, we extend the results of Section 3 to this new model[1].

2 Preliminaries

In this section, the main concepts and notations are defined. More definitions will be given in the following sections. the reader is referred to [3] for any undefined concept.

2.1 The basic model.

The basic structures from which the plant will be built are Finite State Machines (FSMs) [3], that are defined by a 5-tuple $\langle \Sigma, \mathcal{X}, \mathcal{X}_o, \mathcal{X}_f, \delta \rangle$, where Σ is the finite alphabet of G. $\mathcal{X}$ is the finite set of states, $\mathcal{X}_o \subseteq \mathcal{X}$ is the set of initial states, whereas $\mathcal{X}_f$ is the set of final (marked) states of G, δ is the partial transition function defined over $\Sigma \times \mathcal{X} \longrightarrow \mathcal{X}$. The notation $\delta(\sigma, x)!$ means that $\delta(\sigma, x)$ is defined, i.e., there is a transition labeled by an event σ out of state x in machine G. Likewise, $\delta(s, x)$ denotes the state reached by taking the sequence of events defined by trace s from state x in machine G. $\delta(x)$ denotes the active event set of x. Similarly, $\delta^{-1}(x)$ denotes the set of events that lead to x. The behavior of the system is described by a pair of languages $\mathcal{L}(G) \subseteq \Sigma^*$ and $\mathcal{L}_m(G)$. $\mathcal{L}(G)$ is the language generated by G. Similarly, $\mathcal{L}_m(G)$ corresponds to the marked behavior of the FSM G, i.e., the set of trajectories of the system ending in $x_f \in \mathcal{X}_f$. An FSM G is said to be *blocking* if $\mathcal{L}(G) \neq \overline{\mathcal{L}_m(G)}$ and non-blocking if $\mathcal{L}(G) = \overline{\mathcal{L}_m(G)}$, where $\overline{K}$ denotes the prefix closure of the

[1] Proofs can be found in an extended version available at http://www.irisa.fr/vertecs/Publis/Ps/2003-ECC-Extended-version.pdf.

language K. It can be shown [3] that G is non-blocking whenever it is *trim* with respect to $\mathcal{X}_o$ and $\mathcal{X}_f$ (i.e., all the states of G are reachable from $\mathcal{X}_o$ and co-reachable to $\mathcal{X}_f$). We now introduce the notion of submachines of an FSM [3]. An FSM $H = \langle \Sigma_H, \mathcal{X}_H, \mathcal{X}_{o_H}, \mathcal{X}_{f_H}, \delta_H \rangle$ is a submachine of G, denoted $H \subseteq G$, if $\Sigma_H \subseteq \Sigma$, $\mathcal{X}_H \subseteq \mathcal{X}$, $\mathcal{X}_{o_H} \subseteq \mathcal{X}_o$, $\mathcal{X}_{f_H} \subseteq \mathcal{X}_f$, $\forall \sigma \in \Sigma_H, x \in \mathcal{X}_H \ \ \delta_H(\sigma, x)! \Rightarrow (\delta_H(\sigma, x) = \delta(\sigma, x))$.

2.2 Review of the Supervisory Control Problem

Supervisory control theory deals with control of Discrete Event Systems. In practice, one of the main control problem is the invariance problem (or dually the state avoidance control problem), i.e. the supervisor has to control the plant so that the controlled plant remains in a safe set of states (or dually do not reach a set of forbidden states). Throughout the remainder of this paper, we will focus on the latter interpretation. Assume a plant G is given and modeled as an FSM and a set of states E, we recall how to synthesize a supervisor that will ensure the avoidance of a given set of states E. Knowing that some events are uncontrollable Σ_{uc}, as opposed to the set of controllable events (Σ_c), we first recall the definition of a *controllable submachine* [9].

Definition 1 *Let G be a FSM and H be a submachine of G, then H is controllable w.r.t. G and Σ_{uc}, whenever*

$$\forall x \in \mathcal{X}_H \subseteq \mathcal{X}, \forall \sigma \in \Sigma_{uc}, \delta(\sigma, x)! \Rightarrow \delta_H(\sigma, x)!.$$

A supervisor $\mathcal{S} = (S, \mathcal{X}'_o)$ is given by a function $S : \mathcal{X} \rightarrow 2^{\Sigma_c}$, delivering the set of actions that are disabled in state x of G by the control[2], and the new set of valid initial states $\mathcal{X}'_o \subseteq \mathcal{X}_o$ (it could be the case, that in order to ensure an objective, we need to reduce $\mathcal{X}_o$). Write $\mathcal{S}/G$ for the system, consisting of the initial plant G controlled by the supervisor $\mathcal{S}$.

The Basic Supervisory Control Problem *(BSCP)* is then the following: *given G and E a set of states, the problem is to build a supervisor $\mathcal{S}$ such that (1) $\mathcal{S}/G$ is controllable (2) the traversed states do not belong to E and (3) $\mathcal{S}/G$ is maximal(i.e $\forall S'$, $\mathcal{L}(\mathcal{S}'/G) \subseteq \mathcal{L}(\mathcal{S}/G)$).*

In order to compute such a supervisor, we classically introduce two sets of states: the *weak forbidden set* of states and the *border set* that will be used in the remainder of this paper.

Definition 2 *Given an FSM $G = \langle \Sigma, \mathcal{X}, \mathcal{X}_o, \mathcal{X}_f, \delta \rangle$, and $E \subseteq \mathcal{X}$, we denote by $\mathcal{I}(E)$ and $\mathcal{F}(E)$ the weak forbidden states and the border set of E, that are formally defined by:*

$$\mathcal{I}(E) = \{x \in \mathcal{X} \mid \exists s \in \Sigma^*_{uc}, \delta(s, x) \in E\} \quad (1)$$

$$\mathcal{F}(E) = \{x \in \mathcal{X} \setminus \mathcal{I}(E) \mid \exists \sigma \in \Sigma, s.t.\ \delta(\sigma, x) \in \mathcal{I}(E)\} \quad (2)$$

$\mathcal{I}(E)$ corresponds to the set of states from which it is possible to evolve into E by a trace of uncontrollable events, whereas $\mathcal{F}(E)$ corresponds to the set of states from which it is still possible to perform a control on G before evolving into $\mathcal{I}(E)$.

[2] In a more general framework, S is a function from $\mathcal{L}(G)$ into 2^{Σ}.

Proposition 1 *Given an FSM G and $E \subseteq \mathcal{X}$, a set of states to be forbidden by control, the supervisor $\mathcal{S}$ of G given by the pair $(S, \mathcal{X}'_o)$, such that $\forall x \in \mathcal{X}$*

$$\begin{array}{rcl} S(x) & = & \begin{cases} \{\sigma \in \Sigma_c \mid \delta(x, \sigma) \in \mathcal{I}(E)\} \text{ if } x \in \mathcal{F}(E) \\ \emptyset \text{ Otherwise} \end{cases} \\ \mathcal{X}'_o & = & \mathcal{X}_o \setminus \mathcal{I}(E) \end{array} \quad (3)$$

ensures the invariance of $\mathcal{X} \setminus E$ in G and is maximal.

If $\mathcal{X}'_o = \emptyset$, then it means that the BSCP has no solution. In this case, the obtained Supervisor $\mathcal{S} = (S, \mathcal{X}'_o)$ will be called the trivial supervisor. This notion will be useful in the next section.

The Non-blocking Supervisory Control problem. In most situations, it is of interest to avoid blocking in the resulting controlled system. We then define the notion of *non-blocking supervisor* as one which will allow the controlled closed-loop system to properly terminate in one of its final states. In other words, the *Non-Blocking Supervisory Control Problem* is now the following:

Given G and E a set of states, the problem is to build a supervisor $\mathcal{S}$ such that (1) $\mathcal{S}/G$ is controllable w.r.t. G and Σ_{uc} (2) all the trajectories of $\mathcal{S}/G$ eventually lead to $\mathcal{X}_f$ (3) the traversed configurations do not belong to E and (4) $\mathcal{S}/G$ is maximal.

The standard algorithm that computes the greatest non-blocking controlled system is an iterative algorithm starting with G. The iterative procedure consists of (i) removing the weak forbidden states of E, i.e. $\mathcal{I}(E)$, and (ii) removing states that are not reachable and co-reachable. Call $G^{\uparrow} \subseteq G$ the result. If $G^{\uparrow}$ is not reduced to the empty FSM, then $G^{\uparrow}$ is the greatest controllable submachine of G that both ensures the invariance of $\mathcal{X} \setminus E$ and the reachability of $\mathcal{X}_f$. In the sequel we will refer as $\mathcal{S}^{\uparrow}$ for the resulting supervisor and we will say that $\mathcal{S}^{\uparrow}$ is a non-blocking supervisor whenever, the close-loop system is non-blocking.

3 Control of asynchronous FSMs

In this section, we are interested in a plant G modeled as a collection of FSMs $(G_i)_{i \leq n}$, that behave asynchronously. After a presentation of the model, we first present the (modular) state avoidance control problem for such plants, and then give a sufficient condition under which the obtained supervisor is non-blocking.

3.1 The Model

The plant G, we now consider is modeled as a collection of FSMs $(G_i)_{i \leq n}$, that behave asynchronously (i.e. $\Sigma_i \cap \Sigma_j = \emptyset$, $i \neq j$). Such plants are called product plants in [4]. The global behavior of G is then given by the FSM $G_1 \parallel \cdots \parallel G_n$, where the operation $\parallel$ is the classical shuffle operation performing the beahvioral interleaving. Throughout the remainder of this section, the different components of the plant are designed to be non-blocking (i.e. each FSM G_i is assumed to be trim). This clearly entails that $G = G_1 \parallel \cdots \parallel G_n$ is also non-blocking.

Remark that, as each component of G is assumed not to have interaction with the other. However, as we will see in the next sections, interactions between the various components G_i of the system will be made by means of a supervisor that will allow or not events to be triggered in the different components of G. In other words, a supervisor will be used to coordinate the evolution between the components.

Notations. Following the $\|$ definition, states of the plant G are of the form $\langle x_1, \ldots, x_n \rangle$, where each $x_i \in \mathcal{X}_i$ of G_i. For convenience, we will call a state, any element $x_i \in \mathcal{X}_i$ of a particular FSM G_i, and a *configuration* a tuple of the form $\langle x_1, \ldots, x_n \rangle$ of G, i.e. a "state" of the resulting FSM. Moreover we call $\mathcal{X}^F$ the set of configurations of G. Note that, as $\forall i \leq n, G_i$ is assumed to be trim, we have $\mathcal{X}^F = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$.

3.2 The Basic State Avoidance Control Problem (BSACP)

Let us now consider a plant modeled as a collection of asynchronous FSMs $(G_i)_{i \leq n}$. Our aim is to solve the state avoidance control problem for a set of forbidden configurations of the form $E = E_1 \times \cdots \times E_n$, i.e. all the configurations of G that belongs to the set $\{\langle x_1, \ldots, x_n \rangle \in \mathcal{X}^F \mid \forall i \leq n,\ x_i \in E_i\}$ are forbidden (in the sequel, such a set of configurations will be called a cube). We provide a method that solves locally the control problem (i.e. on each component of G without computing the whole system) but produces a global supervisor ensuring the invariance of $\mathcal{X}^F \setminus E$.

Proposition 2 *Assume given n FSMs G_i of the form $\langle \Sigma_i, \mathcal{X}_i, \mathcal{X}_{o_i}, \mathcal{X}_{f_i}, \delta_i \rangle$, $i = 1, \ldots, n$ and n subsets of states $(E_i)_{i \leq n}$. Consider the sets $\mathcal{F}(E_i)$, $\mathcal{I}(E_i)$ as defined in Definition 2 as well as the corresponding supervisors $\mathcal{S}_i$.*

Let $\mathcal{S}_E = (S, \mathcal{X}_{o_E})$ be such that $\forall x = \langle x_1, \ldots, x_n \rangle$,

$$S_E(x) = \bigcup_{1 \leq i \leq n} \{S_i(x_i) \mid x_i \in \mathcal{F}(E_i) \text{ and } \forall j \neq i,\ x_j \in \mathcal{I}(E_j)\}$$

$$\mathcal{X}_{o_E} = \{\langle x_{o_1}, \ldots, x_{o_n} \rangle \in \Pi_{i \leq n} \mathcal{X}_{o_i} \ / \ \exists i,\ x_{o_i} \in \mathcal{X}'_{o_i}\}$$

The supervisor $\mathcal{S}$ ensures the avoidance of $E = E_1 \times \ldots \times E_n$. in $G = G_1 \parallel \cdots \parallel G_n$ and is maximal. Moreover, if there exists at least one supervisor $\mathcal{S}_i$ that is not trivial, then $\mathcal{S}$ is not trivial.

let S_E be a supervisor as in Proposition 2, one can see that at most one supervisor is active at a time. It is the one for which the sub-plant has evolved in its border set of states when the other sub-plants are in a forbidden state.
The interest of such a method is that the supervisor $\mathcal{S}_E$ is locally computed according to the local supervisors $\mathcal{S}_i$. Therefore, this method avoids the building of the whole system and the computation of $\mathcal{S}$ on the resulting system. Hence, it reduces the complexity of the algorithm (see the next section) as well as the memory storage of the supervisor. Moreover, the supervisor itself somehow keeps the structure of the plant as it is represented as a collection of local supervisors. To our mind, the way $\mathcal{S}_E$ is built may improve the readability and the understanding of the control effect. Finally, note that this kind of control objectives cannot be solved using the method presented in [4] (the whole system has to be computed).

3.3 Modular BSACP

So far, the considered configuration sets to be forbidden were particular, in the sense that each subset of $\mathcal{X}^F$ can not be represented as it. We now consider a more general set of configurations E of the form $\bigcup_{1 \leq i \leq m} E^i$ where for $1 \leq i \leq m$, E^i is a cube of the form $E^i = E^i_1 \times \cdots \times E^i_n$[3].

Proposition 3 *Let $G = G_1 \parallel \cdots \parallel G_n$ be the plant to be controlled and a set $E = \bigcup_{1 \leq i \leq m} E^i$ where $\forall 1 \leq i \leq m$, $E^i = E^i_1 \times \cdots \times E^i_n$ and $E^i_j \subseteq \mathcal{X}_j$ for $1 \leq j \leq n$. Let $\mathcal{S}_{E^i} = (S_{E^i}, \mathcal{X}_{o_{E^i}})$ be the supervisors computed w.r.t. G and E^i, then $\mathcal{S}_E = (S, \mathcal{X}_{o_E})$, where $\forall x = \langle x_1, \ldots, x_n \rangle \in \mathcal{X}^F$,*

$$\begin{cases} S(x) = S_{E^1}(x) \cup \cdots \cup S_{E^m}(x) \\ \mathcal{X}_{o_E} = \mathcal{X}_{o_{E^1}} \cap \cdots \cap \mathcal{X}_{o_{E^m}} \end{cases} \quad (4)$$

ensures the invariance of $\mathcal{X} \setminus E$ in G and is maximal.

Let us now discuss about complexity of the control synthesis phase. Given an FSM with N states and a set of states to be avoided by control, assume that the complexity of this controller synthesis phase is in $\mathcal{O}(f(N))$. Let us now consider a system G of the form $G_1 \parallel \cdots \parallel G_n$ where each G_j contains N states. Due to the asynchronous and the trim assumptions, the number of states of G is N^n. Hence, using classical techniques, the state avoidance control problem is in $\mathcal{O}(f(N^n))$. In our case, given a set of forbidden configurations E composed of m cubes, then the offline supervisor computation complexity is in $\mathcal{O}(m.n.f(N))$. However one have also to take into account the computations that have to be done on-line when controlling the plant. Indeed, deciding which supervisor have to be activated given one configuration, is done at execution time. This can be done in $\mathcal{O}(m.n.N)$.

3.4 Non-Blocking Supervisory Control Problem

In the previous section, it may happen that the resulting controlled system be blocking. Assume given a plant G of the form $G_1 \parallel \cdots \parallel G_n$ and a forbidden cube $E = E_1 \times \cdots \times E_n$, we now give a sufficient condition for the controlled system obtained using the methodology of Section 3.2 to be non-blocking.

Proposition 4 *Let $\mathcal{S}_E$ be the supervisor computed as in Proposition 2 w.r.t. G and E. Then, if $\forall 1 \leq i \leq n$,*

(a) *either $\forall x_i \in \mathcal{X}_i \setminus \mathcal{I}(E_i)$, $\exists x_{f_i} \in \mathcal{X}_{f_i} \setminus \mathcal{I}(E_i)$ that is reachable from x_i in S_{E_i}/G_i*

(b) *or $\exists j \neq i$, $\forall x_j \in \mathcal{X}_j$, $\exists x_{f_j} \in \mathcal{X}_{f_j} \setminus \mathcal{I}(E_j)$ that is reachable from x_j in G_j*

then $\mathcal{S}_E$ is non blocking. Moreover, $\mathcal{S}_E = \mathcal{S}^{\uparrow}$, where $\mathcal{S}^{\uparrow}$ is computed on G as in Section 2.2 w.r.t.E.

[3] Note that any subset of $\mathcal{X}^F$ can be represented as a union of cubes.

What the above property states is that the obtained supervisor is blocking whenever there exists a state x_i in an FSM G_i such that the final states of G_i are not reachable from x_i under the control of the local controller $\mathcal{S}_{E_i}$ and for all the other FSMs all the final states belong to the weak forbidden set of states. Obviously, $\mathcal{S}_E$ is blocking only under strong hypothesis. Nevertheless, if these conditions do not occur, then one has to find a way to avoid the blocking configurations. This aspect is currently under investigation.

4 The Hierarchical Finite State Machine

So far, we gave results dealing with the control of asynchronous product of FSMs. We now extend these results to the case of Hierarchical Finite State Machines (HFSM). A Hierarchical Finite State Machines is an FSM which includes new features like the nesting of state machines (inducing the hierarchy) and the re-usability of components. From now on, some states (called super-states) of an FSM can be other FSMs. Informally, the meaning of such a hierarchical definition is obtained by substituting each super-state by a set of asynchronous FSMs running in parallel. Such a model is called Hierarchical Finite State Machine (HFSM). We hereby focus on a two-level Finite State Machine, knowing that the results presented in the next section can be extended to a multi-level hierarchical finite state machine (See [8] for a more complete review of the HFSMs).

4.1 Definition of an HFSM

To take into account the hierarchy, we need to introduce the notion of structure which represents the upper level of the HFSM.

Definition 3 *A structure K is a tuple $\langle \Sigma, \mathcal{X}, \mathcal{B}, \mathcal{X}_o, \mathcal{X}_f, \delta \rangle$, where $\mathcal{X}$ is a set of atomic states, $\mathcal{X}_o \subseteq \mathcal{X}$ is the set of initial states and $\mathcal{X}_f \subseteq \mathcal{X}$ is the set of final states. $\mathcal{B}$ is the set of super-states of K. δ is the partial transition function of K defined over $\Sigma \times \{\mathcal{X} \cup \mathcal{B}\} \rightarrow \{\mathcal{X} \cup \mathcal{B}\}$.* •

In the following we will denote by $K^A = \langle \Sigma, \mathcal{X} \cup \mathcal{B}, \mathcal{X}_o, \mathcal{X}_f, \delta \rangle$ the structure K seen as an FSM (i.e. when the super-states are considered as atomic states).

Definition 4 *A Hierarchical Finite State Machine $\mathcal{K}$ is given by a tuple $(\langle K, G_1, \ldots, G_n \rangle, Y, I)$, where K is a structure as defined in Def 3 and $\forall 1 \leq i \leq n$, $G_i = \langle \Sigma_i, \mathcal{X}_i, \mathcal{X}_{o_i}, x_{f_i}, \delta_i \rangle$ is an FSM, and Y, I are two functions that characterize the hierarchy and the composition between the FSMs.*

- *$Y : \mathcal{B} \longrightarrow 2^{\langle G_1, \ldots, G_n \rangle}$ is a function which maps each super-state $b \in \mathcal{B}$ on a set of FSMs G_i, with $i \geq 1$. We use J_b as $\{j \leq n |\ G_j \in Y(b)\}$. The structures of $Y(b)$ behave asynchronously.*
- *I is an Input function that gives access to the set of initial states that are reached when triggering an event that makes the system evolve into a super-state b. $\forall b \in \mathcal{B}$, $I(b)$ is a function defined over $\Pi_{j \in J_b} \mathcal{X}_{o_j} \rightarrow 2^{\Sigma_i}$. Given a super-state $b \in \mathcal{B}$, and $x_o = \langle x_{o_1}, \ldots, x_{o_{\|J_b\|}} \rangle$ a tuple of initial states, $I(b)(x_o)$ corresponds to the events that make the system go from its current state into x_o.* •

K will be called the *root* of $\mathcal{K}$, whereas the FSMs G_i will be called the *leaves* of $\mathcal{K}$.

Assumptions. In order to be able to perform control on HFSM, we need to make some assumptions on it:

1. $\forall i, j$, s.t. $\exists b \in \mathcal{B},\ G_i,\ G_j \in Y(b),\ \Sigma_i \cap \Sigma_j = \emptyset$ (*asynchrony of parallel FSMs*).
2. Let $b \in \mathcal{B}$ and let $(G_j)_{j \in J_b} = Y(b)$ be the corresponding FSMs attached to b. Then, the set $(I(b)(x_o))_{x_o \in \Pi_{j \in J_b}(\mathcal{X}_{o_j})}$ is a partition of $\delta^{-1}(b)$ (*entering a super-state is deterministic*).

The behavior of $\mathcal{K}$. Let $\mathcal{K} = (\langle K, G_1, \ldots, G_n \rangle, Y, I)$ be an HFSM. $\mathcal{K}$ is initialized in one of the initial states of K and as long as no super-state is reached, the behavior of $\mathcal{K}$ corresponds to the one of the FSM K^A. Assume now that the HFSM is in a state x such that $\delta(\sigma, x) = b \in \mathcal{B}$ and that σ is triggered. Then all the structures of $Y(b)$ are simultaneously activated and entered in one of their initial states according to $I(b)$, i.e. $\mathcal{K}$ in the configuration $x_o = \langle x_{o_{j_1}}, \ldots, x_{o_{j_{\|J_b\|}}} \rangle$, such that $\sigma \in I(b)(x_o)$. Further, the different structures evolve asynchronously. However, in order to evolve out of a super-state b, there is a synchronization between the different structures of $Y(b) = (G_i)_{i \in J_b}$ on their final state. Hence, an event $\sigma \in \delta(b)$ can be triggered in a super-state b whenever each substructure of $Y(b)$ is in its corresponding final state (i.e., there is no preemption); the output of a super-state is synchronized with the end of each of the tasks associated with the different structures involved in this super-state).

Given a HFSM $\mathcal{K} = (\langle K, G_1, \ldots, G_n \rangle, Y, I)$, we can make correspond an FSM which is obtained by replacing each super-state b by its corresponding FSM K_b^F obtained by performing the asynchronous product between each FSM of $Y(b)$ (the initial states of K_b^F are connected to the states of K according to I (resp. for the final state). The result is an FSM, denoted by $\mathcal{K}^F$. Such an FSM is called the *expanded structure* of $\mathcal{K}$.

States and configurations. Due to the hierarchical description of the system, the states of the obtained expanded structure $\mathcal{K}^F$ can have different forms. They are either atomic states or of the form $[b, \langle x_1, \ldots, x_{j_{\|J_b\|}} \rangle]$, with $\{j_1, \ldots, j_{\|J_b\|}\} = J_b$, which intuitively correspond to particular configurations of the HFSM, i.e. the states in which the FSMs are simultaneously at a given instant. In the sequel, we will denote by $\mathcal{X}^F$ the set of configurations of $\mathcal{K}^F$, i.e. the states of the expanded HFSM.

4.2 The BASCP

In this section, we consider the configuration avoidance problem, namely how to avoid the system to reach some particular configurations during its evolution. The sets of configurations we will consider are as follows:

Forbidden configurations: With the notations of Definition 4, let $\mathcal{K} = (\langle K, G_1, \ldots, G_n \rangle, Y, I)$ be an HFSM, such that $K = (\Sigma, \mathcal{X}, \mathcal{B}, \mathcal{X}_o, \mathcal{X}_f, \delta)$. Given $b \in \mathcal{B}$, $E^b = \bigcup_{1 \leq j \leq m_b} E^{b,j}$ with $E^{b,j} = E^{b,j}_{j_1} \times \cdots \times E^{b,j}_{j_{\|J_b\|}}$ et $E^{b,j}_{j_i} \subseteq \mathcal{X}_{j_i}$ pour $j_i \in J_b$. For

simplicity, the set of configurations $[b, \langle x_{j_1}, \ldots, x_{j_{\|J_b\|}}\rangle]$ such that $\langle x_{j_1}, \ldots, x_{j_{\|J_b\|}}\rangle \in E^b$ is denoted $[b, E^b]$. Now, one can see that every set of configurations of $\mathcal{K}^F$ can be represented by a set E of $\mathcal{K}$ of the form

$$E = E_0 \bigcup \left(\bigcup_{b \in \mathcal{B}} [b, E^b] \right) \quad (5)$$

where $E_0 \subseteq \mathcal{X}$. This set represents the forbidden configurations at the higher level of $\mathcal{K}$, whereas $[b, E^b]$ corresponds to the forbidden configurations at the lower level (i.e. inside the super state b). As in the case of asynchronous FSMs, the idea of the control is to compute supervisors separately for each structure/FSMs (without expanding the system), and then to build a global supervisor, which can be seen as an oracle.

Control of a structure First, we need to extend the definition of weak forbidden set of states introduced in Definition 2 in order to take into account the super-states. The idea is that we do not want to remove a super-state by control as there possibly exists a way to control the system inside this super-state. *A contrario*, let $b \in \mathcal{B}$ a super-state of K, given a control objective, it may happen that we need to restrict the entering in a super-state. Hence, for $A \subseteq \delta^{-1}(b)$, we introduce $b_{|A}$ the "controlled super-state" b considering it is only reachable by triggering an event of A and we denote by $\mathcal{B}_{|\mathcal{A}} = \{b_{|A} \mid b \in \mathcal{B} \text{ and } A \subseteq \delta^{-1}(b)\}$, the corresponding set of controlled super-states. This kind of states are introduced in order to partially forbid some super-states. Indeed, one can only want some super states b to be reachable with respect to a subset of $I(b)(.)$. In fact, this is a way to avoid some initial states of b to be reachable at the lower level of the hierarchy.

Based on these remarks and definitions, we extend the definition of weak forbidden set of states introduced in Definition 2:

Definition 5 *Let $K = \langle \Sigma, \mathcal{X}, \mathcal{B}, \mathcal{X}_o, \mathcal{X}_f, \delta\rangle$ be the root of an HFSM $\mathcal{K}$ and $e \in \mathcal{X} \cup \mathcal{B}_{|\mathcal{A}}$.*

• *If $e \in \mathcal{X}$, then*

$$\mathcal{I}(e) = \{x \in \mathcal{X} \mid \exists s \in \Sigma_{uc}^*,\ \delta(s,x) = e \text{ and } \forall s' \leq s,\ \delta(s',x) \notin \mathcal{B}\}.$$

$$\mathcal{I}_B(e) = \{b \in \mathcal{B} \mid \exists \sigma \in \Sigma_{uc},\ \delta(\sigma, b) \in \mathcal{I}(e)\}$$

• *If $e = b_{|A} \in \mathcal{B}_{|\mathcal{A}}$, then*

$$\mathcal{I}(b_{|A}) = \{x \in \mathcal{X} \mid \exists s \in \Sigma_{uc}^*,\ \exists \sigma \in \Sigma_{uc} \cap A,\ \delta(s\sigma, x) = b \text{ and } \forall s' \leq s,\ \delta_K(s', x) \notin \mathcal{B}\}$$

$$\mathcal{I}_B(b_{|A}) = \{b' \in \mathcal{B} \mid \exists \sigma \in \Sigma_{uc},\ (\delta_K(\sigma, b') \in \mathcal{I}(b_{|A})) \text{ or } (\sigma \in \Sigma_{uc} \cap A \text{ and } \delta(\sigma, b') = b)\}$$

Finally, given $E \subseteq \mathcal{X} \cup \mathcal{B}_{|\mathcal{A}}$,

$$\mathcal{I}(E) = \cup_{e \in E} \mathcal{I}(e),\ \text{and}\ \mathcal{I}_B(E) = \cup_{e \in E} \mathcal{I}_B(e)$$

Intuitively speaking, given $e \in E$, if $e \in \mathcal{X}$, $\mathcal{I}(e)$ (resp. $\mathcal{I}_B(e)$) represents the set of atomic states (resp. super-states) of K from which e can be reached via an uncontrollable trajectory that only traverses atomic states. if $e = b_{|A} \in \mathcal{B}_{|\mathcal{A}}$, the meaning of $\mathcal{I}(e)$ and $\mathcal{I}_B(e)$ is similar except that we ask the last event of the uncontrollable trajectories to belong to A.

The next operator Φ will be useful to compute the set of weak forbidden configurations by going-up/down in the hierarchy of the plant. Indeed, if an initial state of a super-state has to be forbidden by control, then at the higher level, a supervisor has to avoid the system to enter the super-state via this initial state. Conversely, the final state of a super-state that may lead to a forbidden configuration via an uncontrollable trajectory has also to be forbidden by control. This is captured by the definition 6:

Definition 6 *Let $e \in \mathcal{X}^F \cup \mathcal{B}_{|\mathcal{A}}$, , we denote by $\mathcal{X}_{o_b}$ (resp. x_{f_b}) the set of initial states (resp. the final state) of the FSM associated to b (i.e. $\|_{G_i \in Y(b)} G_i$). Now, we define $\Phi(e)$ as follows:*

1. *If $e \in \mathcal{X} \cup \mathcal{B}_{|\mathcal{A}}$, then $\Phi(e) = \mathcal{I}(e) \cup \{[b, x_{f_b}] \mid b \in \mathcal{I}_{\mathcal{B}}(e)\}$*[4]
2. *If $e = [b, \langle x_{j_1}, \ldots, x_{j_{\|J_b\|}}\rangle]$, then given the set $\mathcal{I} = \mathcal{I}(x_1) \times .. \times \mathcal{I}(x_{j_{\|J_b\|}})$*

$$\Phi(e) = \bigcup_{x \in \mathcal{I}} [b, x] cup b_{|\bigcup_{x_{o_b} \in \mathcal{I} \cap \mathcal{X}_{ob}} \{I(b)(x_{o_b})\}}$$

Now, given a set $E \subseteq \mathcal{X}^F \cup \mathcal{B}_{|\mathcal{A}}$, $\Phi(E) = \bigcup_{e \in E} \Phi(e)$.

Given $e \in \mathcal{X}^F \cup \mathcal{B}_{|\mathcal{A}}$, if $e \in \mathcal{X} \cup \mathcal{B}_{|\mathcal{A}}$, then $\Phi(e)$ corresponds to the set of weak forbidden configurations, to which we add the final states of the super-states that can lead in an uncontrollable way into e. This way we are going down in the hierarchy. (point *1.*). Now, if e is a configuration of the form $[b, \langle x_{j_1}, \ldots, x_{j_b}\rangle]$, then $\Phi(e)$ corresponds to the set of weak forbidden configurations inside the super-state b, as well as the restricted super-state itself if some initial states belong to the weak forbidden set of states. Remark that in point *2.* the computations are made locally on each G_i that are involved in the super-states b.

The Supervisor computation Assume given a set of forbidden configurations of the form $E = E_0 \bigcup \left(\bigcup_{b \in \mathcal{B}} [b, E^b]\right)$, as defined by Equation 5, then the way the set of weak forbidden configurations is computed by the following fix-point iteration:

$$\begin{cases} \mathcal{I}_o = E \\ \mathcal{I}_{i+1} = \Phi(\mathcal{I}_i(E)) \cup \mathcal{I}_i \end{cases} \quad (6)$$

Let us call $\mathcal{I}_H(E)$ the result of the previous fix-point computation. Now, one can see that the set of forbidden configurations can be reorganized as follows:

$$\mathcal{I}_H(E) = \mathcal{X}' \cup \mathcal{B}'_{|\mathcal{A}} \cup \bigcup_{b \in \mathcal{B}} [b, E'^b], \quad (7)$$

[4] $\mathcal{I}$ is the function defined in Def 5.

where $E'^b = \bigcup_i E'^{b,i}_{j_1} \times \cdots \times E'^{b,i}_{j_{\|J_b\|}}$ and $\mathcal{X}' \subseteq \mathcal{X}$ and $\mathcal{B}'_{|\mathcal{A}} \subseteq \mathcal{B}_{|\mathcal{A}}$. For the super-states, what the above states, is that it is forbidden to enter these super-states through the events that belongs to a set of events A. Moreover, each super-state b can be seen as a collection of asynchronous FSMs for which the set of configurations E'^b has to be forbidden. Note that as E'^b is given by a union of cubes, in order to control the behavior of $\mathcal{K}$, we will use the modular methodology explained in Section 3.3.

Based on the previous remarks, we then have the following property that makes the link with the expanded HFSM. One can shows that the following property holds.

Proposition 5 *$\mathcal{I}_H(E) \setminus \mathcal{B}'_{|\mathcal{A}} = \mathcal{I}(E)$, where $\mathcal{I}(E)$ is computed with respect to $\mathcal{K}^F$ as in Definition 2.*

In other words, the set $\mathcal{I}_H(E)$[5] corresponds to the weak forbidden set of states $\mathcal{I}(E^F)$ of the plant $\mathcal{K}^F$ seen as a flat FSM. However, compare to the classical methods, all the computations has been performed locally and not on the global plant.

Based on the previous decomposition (7) of $\mathcal{I}_H(E)$, a supervisor can be extracted. It is performed as follows:

1. $\forall b \in \mathcal{B}$, we compute $\mathcal{S}_b$ the supervisor $\mathcal{S}_b = (S_b, \mathcal{X}_{o_b})$ that avoid the set of cubes E'^b to be reachable in $\|_{j \in J_b} G_j$ using the methods developed in Section 3.3. Note that we only need to compute the borders. If some initial states are forbidden in the super-states, then this is taken into account as the upper level, since in this case, this super-state (restricted to the events that does not lead into these initial states) belongs to $\mathcal{B}'_{|\mathcal{A}}$.
2. For K, we compute $\mathcal{S}_K = (S_K, \mathcal{X}'_o)$ defined by

$$\begin{aligned} S_K(e) &= \{\sigma \in \Sigma_c | \delta(\sigma, e) \in \mathcal{X}' \vee \delta(\sigma, e) = b \; s.t. \\ &\qquad\qquad b_{|_A} \in \mathcal{B}'_{|\mathcal{A}} \wedge \sigma \in A\} \\ \mathcal{X}'_o &= \mathcal{X}_o \setminus \mathcal{X}' \end{aligned}$$

   Note that as $\mathcal{X}' \cup \mathcal{B}'_{|\mathcal{A}} = \mathcal{I}(\mathcal{X}' \cup \mathcal{B}'_{|\mathcal{A}})$, we only have to compute the border of this set.

Proposition 6 *With the preceding notations, let $\mathcal{S} = \langle \mathcal{S}_K, (\mathcal{S}_b)_{b \in \mathcal{B}} \rangle$, be such that*

$$S_E(e) = \begin{cases} S_K(e) \text{ if } e \in \mathcal{X} \\ S_K(b) \cup S_b(x_{j_1}, \ldots, x_{j_{\|J_b\|}}) \\ \qquad \text{if } e = [b, \langle x_{j_1}, \ldots, x_{j_{\|J_b\|}} \rangle] \\ \emptyset \text{ Otherwise} \end{cases} \quad (8)$$

$$\mathcal{X}_{o_E} = \mathcal{X}'_o$$

Then, the supervisor $\mathcal{S}$ ensures the invariance of $\mathcal{X}^F \setminus E$ and is maximal. $\diamond$

To conclude this section, let us remark that as in Section 3, we made the necessary efforts not to expand the HFSM in order to compute the supervisor. In particular, the set of weak forbidden configurations has been computed locally on each submachine and on the upper level of the HFSM.

[5] to which we removed $\mathcal{B}'_{|\mathcal{A}}$ as, this set actually corresponds to the set of initial states that are forbidden in a super-states. Hence, they are taken into account by some $\langle b, E^b \rangle$.

5 Conclusion & Future Works

In this paper we have considered the control of structured plant modeled as Asynchronous Finite States Machines and Hierarchical Finite State Machines. Based on this model, we proposed an algorithm allowing the computation of a supervisor solving the State avoidance control problem. The control objective is given as a collection of forbidden configurations that we decompose according to the local FSMs. Based on this decomposition, we locally solve the problem with respect to the local FSMs and finally we provide a global supervisor ensuring the global property. As all the computations are done on the sub-plants according to the local specification, there is no need to build global plant, hence reducing the complexity of the supervisor computation. In the case of modular Plant, we gave a sufficient condition under which the resulting controlled plant is non-blocking. We are currently looking for an algorithm that will force the plant to be non-blocking while still avoiding the computation of the whole state space. Another point of interest would be to extend the model by adding preemption and synchronizations between the FSMs, which is obviously one of the main limitation of the presented work. The generalization of the control objectives is also under investigation.

References

[1] Y. Brave and M. Heimann. Control of discrete event systems modeled as hierarchical state machines. *IEEE Transacations on Automatic Control*, 38(12):1803–1819, December 1993.

[2] P. Caines, V. Gupta, and G. Shen. The hierarchical control of ST-Finite State Machines. *Systems & Control Letters*, 1997.

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

[4] M.H. deQueiroz and J.E.R. Cury. Modular supervisory control of large scale discrete-event systems. In *Discrete Event Systems: Analysis and Control. Proc. WODES'00*, pages 103–110. Kluwer Academic, 2000.

[5] P. Gohari-Moghadam. *A linguistic Framework for controller hierarchical DES*. M.a.s.c. thesis, Dept. of Electl. & Comptr. Engrg., University of Toronto, April 1998.

[6] S. Kim, J. Park, and R. Leachman. A supervisory control approach for execution control of an FMC. *The International Journal of Flexible manufacturing Systems*, 13:5–31, 2001.

[7] R.J. Leduc. *Hierarchical Interface Based Supervisory Control*. PhD thesis, Dept. of Elec. & Comp. Engrg., Univ. of Toronto, 2002.

[8] H. Marchand and B. Gaudin. Supervisory control problems of hierarchical finite state machines. In *41th IEEE Conference on Decision and Control*, Las Vegas, USA, December 2002.

[9] P. J. Ramadge and W. M. Wonham. The control of discrete event systems. *Proceedings of the IEEE; Special issue on Dynamics of Discrete Event Systems*, 77(1):81–98, 1989.

[10] K. C. Wong and W. M. Wonham. Hierarchical control of discrete-event systems. *Discrete Event Dynamic Systems*, 6:241–273, 1996.

[11] W. M. Wonham and P. J. Ramadge. Modular supervisory control of discrete event systems. *Mathematics of Control Signals and Systems*, 1:13–30, 1988.