Contrôle de systèmes à événements discrets hiérarchiques

Benoit Gaudin, Hervé Marchand

Contrôle de systèmes à événements discrets hiérarchiques

B. Gaudin — H. Marchand

VerTeCs
IRISA/INRIA Rennes
http://www.irisa.fr/vertecs/
Prénom.Nom@irisa.fr

RÉSUMÉ. *Dans ce papier, nous nous intéressons au contrôle de systèmes à événements discrets modélisés par des machines à états finis hiérarchiques. Le problème du contrôle que nous nous posons est d'assurer l'interdiction d'un ensemble particulier de configurations dans le système. Nous présentons des algorithmes qui, basés sur une décomposition particulière de cet ensemble, résolvent localement les problèmes de contrôle (i.e. sur chaque composant du système sans avoir à calculer explicitement le système) et produisent un contrôleur global assurant la propriété attendue. Ce type d'objectifs peut être utilisé pour décrire/assurer des interactions entre différents sous-systèmes.*

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.*

MOTS-CLÉS : *Synthèse de contrôleurs, modèle hiérarchique et asynchrone, modularité*

KEYWORDS: *Supervisory Control Problem, Hierarchical & Asynchrounous model, M odularity*

1. Introduction

La théorie du contrôle consiste à restreindre le comportement d'un système par le biais d'un superviseur de manière à ce que le système ainsi contrôlé soit correct vis à vis d'un ensemble de propriétés (ou d'objectifs de contrôle) que le système initial ne vérifiait pas. Dans cet article, nous adoptons le formalisme de [RAM 89]. Étant donné un système (P) et une spécification (S) de son comportement attendu, le contrôle du système consiste en l'interdiction de certains événements, appelés événements contrôlables de manière à ce que le comportement du système contrôlé soit inclu dans celui de (S).

Pour la plupart des applications considérées, les systèmes à contrôler sont à l'origine spécifiés de manière hiérarchique et modulaire. En revanche la synthèse s'applique sur le système mis à plat. Sachant que la complexité des algorithmes de calcul croît exponentiellement avec le nombre d'états des systèmes mis en parallèle et imbriqués, il semble pertinent de trouver des algorithmes de synthèse qui réalisent le calcul des superviseurs en tirant avantage de la structure du système sans le mettre à plat (i.e. calculer un superviseur localement sur chaque sous-système et d'inférer un superviseur global assurant l'objectif sur le système à contrôler).

Différentes approches ont été considérées afin de réduire la complexité lors de la phase de synthèse. Dans [WON 88], la méthode consiste à fragmenter l'objectif de contrôle en sous-objectifs et à calculer des superviseurs pour chacun de ces objectifs. Le superviseur global est alors vu comme une union des superviseurs (i.e. un évènement est interdit lorsqu'il est interdit par au moins un des superviseurs assurant les sous-objectifs). Notre approche est différente dans la mesure où c'est le système qui est modulaire et non les objectifs de contrôle. En ce sens, nos travaux se situe plus dans l'esprit de [deQ 00, deQ 02]. En effet dans [deQ 00, deQ 02], les auteurs considèrent des systèmes modélisés par un ensemble de sous-systèmes asynchrones et décomposent les objectifs de contrôle en fonction de ces sous-systèmes de manière à obtenir des superviseurs locaux sur chacun des sous-systèmes. Toutefois, lorsque les objectifs servent à exprimer des interactions entre sous-systèmes, cette méthode est équivalente au calcul du système global (i.e., tout le système se doit d'être construit) du fait que l'objectif ne peut être découplé sur chacun des sous-systèmes. Une approche connexe (mais avec les mêmes limitations) basée sur une méthode de synthèse incrémentale a étés étudiée dans [BRA 00]. Nous donnons en Section 3 une méthode permettant de résoudre efficacement ce problème. L'idée n'est pas tant de calculer des superviseurs décentralisés sur chaque composant pouvant agir de manière indépendante, mais plutôt de calculer localement des superviseurs sur chaque composant et de regrouper l'information donnés par ces superviseurs via un oracle.

D'un autre coté, des techniques basées sur l'agrégation d'états ont été proposées dans e.g. [WON 96] afin de réaliser du contrôle hiérarchique. Notre démarche ici est inverse. Le système est spécifié hiérarchiquement par un modèle simplifié des STATECHARTS [HAR 85], les machines à états finis hiérarchiques (abrégé HSFM pour Hierarchical Finite State Machines). Le modèle que nous considérons peut être caractérisé

par une collection de structures imbriquées $\langle K_1, \ldots, K_n \rangle$, où K_1 représente le plus haut niveau de la HFSM. À un niveau intermédiaire, la structure K_i est une HFSM, pour laquelle les états sont soit des "états ordinaires" soit des "macro-états b" qui sont constitués d'un ensemble de structures $(K_j)_{j \in J_b} \subseteq 2^{\langle K_{i+1}, \ldots, K_n \rangle}$, évoluant en parallèle. Chaque structure peut avoir plusieurs états initiaux (resp. finaux). Le comportement d'une structure est le suivant : lorsque le système transite dans un macro-état b, toutes les structures associées à b sont activées et initialisées dans un de leurs états initiaux. *A contrario*, la sortie d'un macro-état est synchronisée avec la fin des tâches associées aux différentes structures de b (i.e. chaque structure est dans un état final). Entre deux états (ordinaire ou macro), le comportement de la structure est similaire à celui d'un automate classique. En Section 4, nous proposons des algorithmes qui synthétisent des superviseurs assurant l'interdiction d'ensembles d'états du système (cf. [BRA 93, GOH 98, LED 02] pour des travaux connexes). Le superviseur global est déterminé à partir de superviseurs locaux, bien que l'objectif initial porte sur le système global. Ce superviseur peut être vu comme un oracle qui active/déactive les superviseurs locaux en fonction de l'état courant du système. De plus, la structure du superviseur reflète celle du système, permettant ainsi d'améliorer la lisibilité et la compréhension des effets du contrôle, et de minimiser la place mémoire nécessaire pour stocker le superviseur.

L'article est organisé comme suit. Dans la section 2, nous présentons le modèle de base que nous utilisons ainsi que les principes de la synthèse de contrôleur. Le modèle de HFSM que nous considérons se caractérisant entre autres, à un niveau donné de la hiérarchie, par des systèmes mis en parallèle, nous résolvons dans un premier temps en Section 3, un problème de synthèse pour des systèmes modélisés par une collection de machines asynchrones. Puis en Section 4, nous étendons ces résultats au cas de systèmes hiérarchiques.

2. Préliminaires

Dans un premier temps, nous rappelons la description du modèle sur lequel nous allons travailler et introduisons quelques notations. Les Machines à États Finis (abrégé FSM pour *Finite State Machine*) sont utilisées pour modéliser les fragments du système.

Définition 1 *Une FSM G est définie par un quintuplet $\langle \Sigma, \mathcal{X}, \mathcal{X}_o, \mathcal{X}_f, \delta \rangle$, où Σ est l'alphabet fini des actions sur G. $\mathcal{X}$ est l'ensemble fini des états de G, $\mathcal{X}_o \subseteq \mathcal{X}$ est l'ensemble des états initiaux, $\mathcal{X}_f$ représente l'ensemble des états finals (marqués) de G et δ est la fonction partielle de transition définie sur $\Sigma \times \mathcal{X} \longrightarrow \mathcal{X}$.*

Notations : $\delta(\sigma, x)!$ signifie que σ est un événement admissible en x. $\delta(x)$ représente l'ensemble des événements admissibles en x. De plus, $\delta^{-1}(x)$ représente l'ensemble des événements menant en x. $\delta(s, x)$ représente l'état atteint par tirage de la trace s depuis l'état x. Le comportement du système est donné par le langage $\mathcal{L}(G) \subseteq \Sigma^*$.

On définit maintenant le produit asynchrone entre FSMs. Cette opération sera utilisée pour composer plusieurs FSMs impliquées dans la spécification du système global.

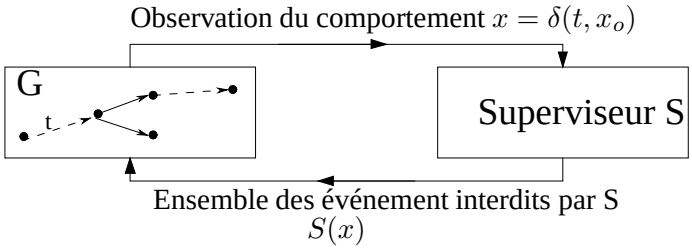
Définition 2 (Composition) *Soient $G_i = \langle \Sigma_i, \mathcal{X}_i, \mathcal{X}_{o_i}, \mathcal{X}_{f_i}, \delta_i \rangle$, $i = 1, 2$ deux FSMs, telles que $\Sigma_1 \cap \Sigma_2 = \emptyset$. Le produit asynchrone de G_1 et G_2, noté $G_1 \| G_2$, est la FSM $\langle \Sigma_{12}, \mathcal{X}_{12}, \mathcal{X}_{o_{12}}, \mathcal{X}_{f_{12}}, \delta_{12} \rangle$, telle que $\Sigma_{12} = \Sigma_1 \cup \Sigma_2$, $\mathcal{X}_{12} = \mathcal{X}_1 \times \mathcal{X}_2$, l'ensemble des états initiaux est donné par $\mathcal{X}_{o_{12}} = \mathcal{X}_{o_1} \times \mathcal{X}_{o_2}$ et l'ensemble des états finals est donné par $\mathcal{X}_{f_{12}} = \mathcal{X}_{f_1} \times \mathcal{X}_{f_2}$. La fonction partielle de transition δ_{12} est définie par :*

$$\delta_{12}(\sigma, \langle x_1, x_2 \rangle) = \begin{cases} \langle \delta_1(\sigma, x_1), x_2 \rangle \text{ si } \sigma \in \Sigma_1 \text{ et } \delta_1(\sigma, x_1)! \\ \langle x_1, \delta_2(\sigma, x_2) \rangle \text{ si } \sigma \in \Sigma_2 \text{ et } \delta_2(\sigma, x_2)! \\ \text{Indéfini sinon} \end{cases} \quad [1]$$

Bref rappel sur la synthèse de contrôleur. Nous rappelons ici brièvement la théorie du contrôle de systèmes à événements discrets de [RAM 89]. Considérons un système G modélisé par une FSM, ainsi qu'un ensemble d'états E de G. L'objectif qui nous intéresse ici est l'interdiction de E, i.e. empêcher les états de E d'être atteignables (sous contrôle).

Parmi les événements de G, il nous faut distinguer les événements sur lesquels un superviseur pourra agir (les contrôlables : Σ_c) de ceux sur lesquels il est impossible d'agir (les incontrôlables : Σ_{uc}).

Superviseur. Le but d'un superviseur est d'agir sur l'évolution du système en permettant (interdisant) l'occurrence d'événements en fonction du comportement passé du système. Formellement, pour un problème d'interdiction d'états, un superviseur $\mathcal{S}$ est un couple $(S, \mathcal{X}'_o)$ tel que S est une fonction $S : \mathcal{X} \rightarrow 2^{\Sigma}$, qui retourne l'ensemble des événements devant être empêchés dans l'état x de G par contrôle[1]. L'ensemble des états initiaux valides par contrôle est représenté par $\mathcal{X}'_o \subseteq \mathcal{X}_o$ (il se peut en effet que $\mathcal{X}_o$ soit réduit afin d'assurer un objectif). On note $\mathcal{S}/G$ le système G contrôlé par $\mathcal{S}$ et $\mathcal{L}(\mathcal{S}/G)$ le langage traduisant son comportement.



Comme tous les événements ne sont pas contrôlables, tous les superviseurs ne sont pas admissibles. En particulier, un superviseur ne doit pas interdire l'occurrence d'événements incontrôlables. Ceci est formalisé par la définition 3 (pour une sous-machine):

Définition 3 *Soit G une FSM et H une sous-machine de G. H est contrôlable par rapport à G et Σ_{uc} si $\forall x \in \mathcal{X}_H \subseteq \mathcal{X}, \forall \sigma \in \Sigma_{uc}, \delta(\sigma, x)! \Rightarrow \delta_H(\sigma, x)!$.*

1. Dans le cadre général, un superviseur est une fonction qui étant donné une séquence passée du système renvoie l'ensemble des événements interdits (C.f. [CAS 99], Chapitre. III pour plus de détails).

Problème Classique de la Supervision *(PCS)* : *étant donné un système G et un ensemble E d'états de G, synthétiser un superviseur $\mathcal{S}$ tel que (1) les états atteignables durant l'exécution n'appartiennent pas à E,(2)$\mathcal{S}/G$ soit contrôlable par rapport à G et Σ_{uc} et (3) $\mathcal{S}/G$ soit maximale (i.e $\forall S'$ assurant l'interdiction de E sur G, $\mathcal{L}(\mathcal{S}'/G) \subseteq \mathcal{L}(\mathcal{S}/G)$).*

Le superviseur résolvant le PCS peut être défini à partir des ensembles *faiblement interdits* et *frontières*, définis par

Définition 4 *Étant donnés une FSM $G = \langle \Sigma, \mathcal{X}, \mathcal{X}_o, \mathcal{X}_f, \delta \rangle$ et un ensemble d'états interdits $E \subseteq \mathcal{X}$, l'*ensemble des états faiblement interdits $\mathcal{I}(E)$ *et l'*ensemble des états frontières $\mathcal{F}(E)$ *de E sont définis par :*

$$\mathcal{I}(E) = \{x \in \mathcal{X} \mid \exists s \in \Sigma_{uc}^*, \delta(s,x) \in E\} \quad [2]$$

$$\mathcal{F}(E) = \{x \in \mathcal{X} \setminus \mathcal{I}(E) \mid \exists \sigma \in \Sigma, t.q\ \delta(\sigma,x) \in \mathcal{I}(E)\} \quad [3]$$

Intuitivement, $\mathcal{I}(E)$ correspond à l'ensemble des états desquels il est possible d'atteindre E via une trace d'événements incontrôlables, alors que $\mathcal{F}(E)$ correspond à l'ensemble des états pour lesquels il est nécessaire et possible d'exercer un contrôle sur G afin d'empêcher l'atteignabilité de $\mathcal{I}(E)$.

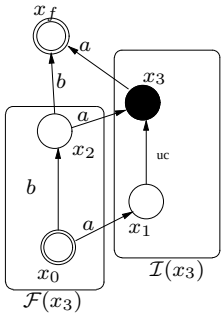
Proposition 1 *Etant donnés une FSM G et un ensemble d'états $E \subseteq \mathcal{X}$, $\mathcal{S} = (S, \mathcal{X}'_o)$, tel que $\forall x \in \mathcal{X}$*

$$\begin{array}{rcl} S(x) & = & \begin{cases} \{\sigma \in \Sigma_c \mid \delta(x,\sigma) \in \mathcal{I}(E)\} \text{ si } x \in \mathcal{F}(E) \\ \emptyset \text{ sinon} \end{cases} \\ \mathcal{X}'_o & = & \mathcal{X}_o \setminus \mathcal{I}(E) \end{array} \quad [4]$$

assure l'interdiction de E dans G et est maximal. ◇

Si $\mathcal{X}'_o = \emptyset$, alors le PCS n'a pas de solution, dans la mesure où tous les états initiaux peuvent mener de manière incontrôlable dans E. Dans ce cas, le superviseur obtenu $\mathcal{S} = (S, \mathcal{X}'_o)$ est dit *trivial*.

Exemple 1 *pour illustrer cette section, considérons la FSM suivante, où l'état x_3 doit être interdit par contrôle. On suppose que $\Sigma_c = \{a, b\}$ et $\Sigma_{uc} = \{uc\}$. L'ensemble des états interdits est donnés par $\mathcal{I}(\{x3\}) = \{x_3, x_1\}$, et l'ensemble des états frontières de $\{x_3\}$ par $\mathcal{F}(\{x_3\}) = \{x_0, x_2\}$.*
Finalement, le superviseur $\mathcal{S}$ est donné par la paire $(S, \{x_o\})$, où S est définie par $S(x_o) = S(x_2) = \{a\}$ et $S(x) = \emptyset$ pour tous les autres états x.



3. Contrôle d'un produit asynchrone de FSMs

Comme expliqué en introduction, avant de décrire le contrôle de machines hiérarchiques, nous allons, dans un premier temps, nous intéresser au contrôle d'un système G modélisé par un produit asynchrone de FSMs $G = G_1 \parallel \cdots \parallel G_n$. Les systèmes tels que G sont particuliers, dans le sens où il n'y a aucune interaction entre les différents sous-systèmes qui le composent. Dans ce cadre, le but d'un superviseur sera de coordonner l'évolution de chacun des ces sous-systèmes les uns par rapport aux autres de manière à ce qu'ils n'évoluent pas dans une configuration dangereuse pour le système global. Prenons l'exemple d'un système composé d'une presse et d'un bras articulé, dont le but est de placer (enlever) un objet dans (de) la presse. Chacun des sous-systèmes peut être modélisé de manière indépendante. De plus, les états du système global tels que "*la presse est fermée*" alors que "*le bras est étendue dans la presse*" sont clairement à éviter au cours de l'exécution du système. Ainsi, dans la pratique, la plupart des objectifs de contrôle seront d'empêcher le système d'atteindre des configurations particulières du système global pouvant se décomposer localement sur chacun des sous-systèmes (dans l'exemple précédent : *la position étendue et dans la presse* pour le sous-système bras et *la position fermée* pour le sous-système presse). Séparément, ils ne correspondent pas à des configurations dangereuses[2]. Ce n'est que lorsque les sous-systèmes se trouvent simultanément dans ces configurations particulières que le système global est dans une situation dangereuse, donc à interdire.

Notations. Avec les notations de la définition 2, les états de G sont de la forme $\langle x_1, \ldots, x_n \rangle$ où $x_i \in \mathcal{X}_i$ pour chaque $1 \leq i \leq n$. Par simplicité, on appelle *état* un élément $x_i \in \mathcal{X}_i$, et *configuration* un état de G. De plus, l'ensemble des configurations atteignables de G sera noté $\mathcal{X}^F = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$.

Résolution du PCS. Considérons un système G modélisé par un produit asynchrone de FSMs $G = G_1 \parallel \cdots \parallel G_n$. On souhaite être capable d'interdire n'importe quel ensemble de configurations de G. Connaissant la structure du système, il paraît assez naturel de spécifier les objectifs de manière structurée. Cet ensemble de configurations sera supposé être de la forme $E = \bigcup_{1 \leq j \leq m} E_1^j \times \cdots \times E_n^j$, permettant ainsi de représenter n'importe quel ensemble de configurations de G.

Pour rendre l'expression du superviseur plus lisible, on s'intéresse dans un premier temps à l'interdiction d'ensemble d'états de la forme $E = E_1 \times \cdots \times E_n$. Un tel ensemble est appelé un *pavé*. La proposition suivante exprime la possibilité de décrire un superviseur assurant l'interdiction de E à partir de superviseurs locaux.

Proposition 2 *Considérons n FSMs $G_i = \langle \Sigma_i, \mathcal{X}_i, \mathcal{X}_{o_i}, \mathcal{X}_{f_i}, \delta_i \rangle$ avec $1 \leq i \leq n$, et un pavé $E = E_1 \times \cdots \times E_n$ t.q. $\forall i \in \{1, \ldots, n\}$, $E_i \subseteq \mathcal{X}_i$. Considérons les ensembles $\mathcal{F}(E_i)$, $\mathcal{I}(E_i)$ (C.f. Déf. 4) ainsi que les superviseurs $\mathcal{S}_i = (S_i, \mathcal{X}'_{o_i})$ correspondants (C.f. Prop. 1).*

2. Il faut noter que ce types d'onjectifs ne peut se décomposer de maniere modulaire sur chacun des composants, ainsi les méthodes développées par [deQ 00] ne peuvent être appliquées efficacement.

Soit $\mathcal{S}_E = (S_E, \mathcal{X}_{o_E})$ *t.q.* $\forall x = \langle x_1, \ldots, x_n \rangle$,

$$S_E(x) = \begin{cases} S_1(x_1) \text{ si } x_1 \in \mathcal{F}(E_1) \text{ et } \forall j \neq 1,\ x_j \in \mathcal{I}(E_j) \\ \ldots \\ S_i(x_i) \text{ si } x_i \in \mathcal{F}(E_i) \text{ et } \forall j \neq i,\ x_j \in \mathcal{I}(E_i) \\ \ldots \\ S_n(x_n) \text{ si } x_n \in \mathcal{F}(E_n) \text{ et } \forall j \neq n \in \mathcal{I}(E_j) \\ \\ \emptyset \text{ Autrement} \end{cases}$$

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

Alors, le superviseur $\mathcal{S}_E$ *assure l'interdiction de l'ensemble* E *dans* $G = G_1 \parallel \cdots \parallel G_n$ *et est maximal. De plus, si il existe un superviseur local* $\mathcal{S}_i$ *qui n'est pas trivial alors* $\mathcal{S}$ *n'est pas trivial.* ◇

Intuitivement, dès lors que tous les composants du système sauf un sont des états localement interdits et que le dernier composant est dans un état frontière alors le superviseur de ce dernier l'empêche d'évoluer dans ses états localement interdits (autrement si un des ces événements était tiré alors le système global évoluerait dans une configuration globalement interdite).

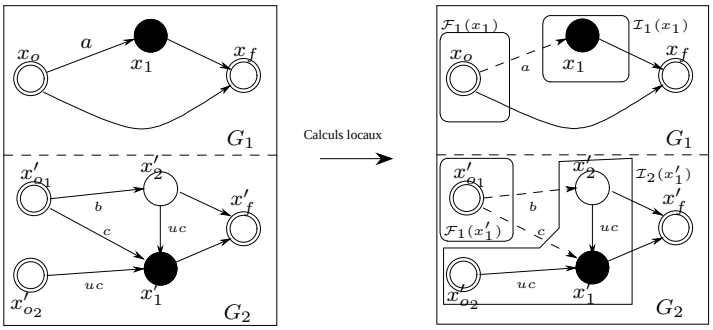
L'intérêt d'une telle méthode est que le superviseur $\mathcal{S}_E$ est déterminé en fonction des superviseurs locaux $\mathcal{S}_i$. La construction du système global peut ainsi être évitée pour calculer un superviseur. La complexité dans ce cas est $\mathcal{O}(n.f(N))$, où n représente le nombre de FSMs impliquées dans le produit, N le nombre d'états de chacune d'elles, et $f(.)$ la complexité de la phase de synthèse d'un superviseur. À contrario, la complexité de la méthode basée sur la construction du système global est en $\mathcal{O}(f(N^n))$. De la même manière, la mémoire nécessaire pour stocker le superviseur obtenu est largement inférieure dans la mesure où il n'est pas nécessaire de calculer globalement le superviseur (seul $\mathcal{F}_i$ et $\mathcal{I}_i$ doivent être calculés et stockés).

Enfin, on peut noter que le superviseur reflète la structure du système initial et que le type d'objectifs de contrôle énoncé ici est plus général que dans [deQ 00] dans la mesure où l'objectif de contrôle n'a pas besoin d'être séparable.

Exemple 2 *Soit* $G = G_1 \parallel G_2$ *(cf. Figure 1(a)). L'objectif est d'éviter la configuration* $x = \langle x_1, x'_1 \rangle$ *d'être atteignable dans* $G_1 \parallel G_2$.

Dans un premier temps, les superviseurs $\mathcal{S}_1$ *(resp.* $\mathcal{S}_2$*) empêchant* x_1 *dans* G_1 *(resp.* x_2 *dans* G_2*) d'être atteignables sont calculés (l'action des superviseurs est représentée par les flèches hachurées sur la figure 1*(b)*). Le superviseur* $\mathcal{S} = (S_{\{x\}}, \mathcal{X}_{o_{\{x\}}})$*, calculé comme en Prop. 2, est tel que* $\mathcal{X}_{o_{\{x\}}} = \{\langle x_o, x'_{o_1} \rangle, \langle x_o, x'_{o_2} \rangle\}$ *et* $S_{\{x\}}(\langle x_o, x'_{o_2} \rangle) = S_{\{x\}}(\langle x_o, x'_1 \rangle) = S_{\{x\}}(\langle x_o, x'_2 \rangle) = \{a\}$ *car* G_1 *est dans* $x_o \in \mathcal{F}(x_1)$ *et* G_2 *est dans* $\mathcal{I}(x'_1)$*.* $S(\langle x_1, x'_{o_1} \rangle) = \{b, c\}$ *(*G_1 *est dans* $\mathcal{I}(x_1)$ *et* G_2 *dans* $\mathcal{F}(x'_1)$*). Pour tous les autres états,* $\mathcal{S}$ *n'interdit aucun événement.* ◇

On va maintenant décrire le superviseur pour le cas général où l'ensemble des configurations que l'on souhaite interdire est de la forme $\bigcup_{1 \leq j \leq m} E^j$ où E^j est un pavé de



(a): $G = G_1 \parallel G_2$ (b): le système contrôlé résultant

Figure 1. *Un exemple simple*

la frome $E^j = E^j_1 \times \cdots \times E^j_n$. On peut remarquer que n'importe quel ensemble de configurations de G peut ainsi être représenté.

Proposition 3 *Soit $G = G_1 \parallel \cdots \parallel G_n$ un système et l'ensemble des états interdits de la forme $E = \bigcup_{1 \leq j \leq m} E^j$ où $\forall 1 \leq j \leq m,\ E^j = E^j_1 \times \cdots \times E^j_n$ et $E^j_i \subseteq \mathcal{X}_i$ pour $1 \leq i \leq n$. Soit $\mathcal{S}_{E^j} = (S_{E^j}, \mathcal{X}_{o_{E^j}})$ les superviseurs assurant l'interdiction de E^j dans G. Le superviseur $\mathcal{S}_E = (S_E, \mathcal{X}_{o_E})$ défini pour tout $x = \langle x_1, \ldots, x_n \rangle \in \mathcal{X}^F$ par*

$$\begin{array}{l} S_E(x) = S_{E^1}(x) \cup \cdots \cup S_{E^m}(x) \\ \mathcal{X}_{o_E} = \mathcal{X}_{o_{E^1}} \cap \cdots \cap \mathcal{X}_{o_{E^m}} \end{array} \quad [5]$$

assure l'interdiction de E et est maximal $\diamond$

L'idée intuitive de cette proposition est que pour calculer le superviseur global assurant l'interdiction de l'ensemble E, il suffit de calculer dans un premier temps les superviseurs assurant l'interdiction des pavés E^j. Par la suite, le superviseur global interdit un événement dès lors qu'il est interdit par un des superviseurs $\mathcal{S}_{E^j}$. Le résultat vient directement des propriétés de [WON 88] sur la synthèse de superviseurs pour des objectifs de contrôle modulaires.

La complexité des calculs des ensembles est en $\mathcal{O}(m.n.f(N))$. Toutefois, il est nécessaire de prendre en compte les calculs à réaliser en ligne. En effet, le choix des superviseurs qui doivent être activés est réalisé durant l'exécution du système contrôlé. Cela peut être effectué en $\mathcal{O}(m.n.N)$, ce qui constitue une complexité acceptable. Toutefois, si les objectives ne sont pas bien structurés, le nombre de pavés nécessaires à la formalisation de l'objectif peut être important. Dans ce cas, les calculs en ligne peuvent s'avérer être coûteux.

4. Le contrôle de Machines à États Finis Hiérarchiques

Précédemment, nous avons donné des résultats concernant le contrôle de systèmes modélisés par un produit asynchrone de FSMs. Nous allons maintenant étendre ces résultats au cas de Machines à États Finis Hiérarchiques (HFSM : Hierarchical Finite State Machines). Nous donnons ici les résultats pour des machines hiérarchiques à deux niveaux (une description plus générale des HFSMs peut se trouver dans [MAR 02]). Intuitivement, une HFSM est une FSM dans laquelle certains des états peuvent être

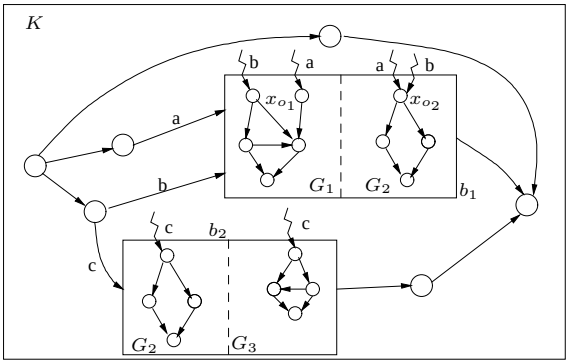


Figure 2. *Un exemple de HFSM à deux niveaux*

raffinés en d'autres FSMs, induisant ainsi une hiérarchie. De plus, certaines de ces FSMs peuvent apparaître à différents endroits et dans différents contextes. Pour illustrer cet aspect, prenons l'exemple d'une cellule flexible de production manufacturière. Les FSMs du niveau inférieur correspondent alors aux différentes machines, alors que la FSM de haut niveau permet d'agencer les machines pouvant être actives simultanément (e.g. G_1 et G_3, où G_2 et G_3 dans l'exemple de la figure 2). Le rôle d'un superviseur sera alors de coordonner/synchroniser les comportements de ces différentes machines de manière à éviter des comportements globaux dangereux.

4.1. *Définition d'une HFSM*

Avant de donner la définition formelle d'une HFSM, nous avons besoin de définir la notion de structure modélisant le comportement haut niveau d'une HFSM.

Définition 5 *Une structure K est un tuple $\langle \Sigma, \mathcal{X}, \mathcal{B}, \mathcal{X}_o, \mathcal{X}_f, \delta \rangle$, où $\mathcal{X}$ est l'ensemble des états atomiques, $\mathcal{X}_o \subseteq \mathcal{X}$ est l'ensemble des états initiaux et $\mathcal{X}_f \subseteq \mathcal{X}$ les états finals. $\mathcal{B}$ est l'ensemble des macro-états de K. δ est la fonction de transition partielle définie sur $\Sigma \times \{\mathcal{X} \cup \mathcal{B}\} \rightarrow \{\mathcal{X} \cup \mathcal{B}\}$.* •

Par la suite K^A désignera la FSM $\langle \Sigma, \mathcal{X} \cup \mathcal{B}, \mathcal{X}_o, \mathcal{X}_f, \delta \rangle$. K^A correspond à la structure K, considérant que tous ses états sont atomiques.

Définition 6 *Une HFSM $\mathcal{K}$ est donnée par un tuple $(\langle K, G_1, \ldots, G_n\rangle, Y, I)$, où K est une structure (C.f. Def. 5) et $\forall 1 \leq i \leq n$, $G_i = \langle \Sigma_i, \mathcal{X}_i, \mathcal{X}_{o_i}, x_{f_i}, \delta_i\rangle$ est une FSM*[3]*. Y, I sont deux fonctions qui caractérisent la hiérarchie et la composition entre les FSMs.*

– $Y : \mathcal{B} \longrightarrow 2^{\langle G_1,\ldots,G_n\rangle}$ est une fonction qui associe à un macro-état $b \in \mathcal{B}$ l'ensemble des FSMs G_i activées en b. On note par J_b l'ensemble des indices $\{j \leq n |\ G_j \in Y(b)\}$. On suppose que $\forall i, j$, t.q. $\exists b \in \mathcal{B},\ G_i,\ G_j \in Y(b),\ \Sigma_i \cap \Sigma_j = \emptyset$.

– I est une fonction telle que $\forall b \in \mathcal{B}$, $I(b)$ est une fonction définie sur $\Pi_{j\in J_b}\mathcal{X}_{o_j} \rightarrow 2^{\Sigma}$. Étant donné un macro-état b et $x_o \in \Pi_{j\in J_b}\mathcal{X}_{o_j}$ un tuple d'états initiaux, $I(b)(x_o)$ est l'ensemble des événements admissibles qui font évoluer le système de son état courant dans x_o. •

Un exemple de HFSM est donné en Figure 2 (par exemple $Y(b_1) = \{G_1, G_2\}$ et $I(b_1)(\langle x_{o_1}, x_{o_2}\rangle) = \{b\}$).

Le comportement de $\mathcal{K}$. Soit $\mathcal{K} = (\langle K, G_1, \ldots, G_n\rangle, Y, I)$ une HFSM. $\mathcal{K}$ est initialisée en un état initial de K et tant qu'aucun macro-état de K n'est atteint, $\mathcal{K}$ se comporte comme la FSM K^A. Soit $b \in \mathcal{B}$ un macro-état de $\mathcal{K}$. Soit $x \in \mathcal{X} \cup \mathcal{B}$, t.q. $\delta(\sigma, x) = b$. On suppose maintenant que $\mathcal{K}$ se trouve dans l'état x. Lorsque σ est tiré, $\mathcal{K}$ évolue vers la configuration $x_o = \langle x_{o_{j_1}}, \ldots, x_{o_{j_{\|J_b\|}}}\rangle$, tel que $\sigma \in I(b)(x_o)$ et se comporte par la suite comme le produit asynchrone $\|_{G_i\in Y(b)}\ G_i$ initialisé en $x_o = \langle x_{o_{j_1}}, \ldots, x_{o_{j_{\|J_b\|}}}\rangle$. Pour "quitter" le macro-état b, chaque FSM de $Y(b)$ doit avoir évolué dans son propre état final. À cet instant, les événements de $\delta(b)$ sont admissibles. En d'autres termes, il n'y a pas de préemption et la sortie d'un macro-état est synchronisée avec la fin des taches associées aux différentes FSMs de ce macro-état.

Notons que les comportements d'une HFSM $\mathcal{K} = (\langle K, G_1, \ldots, G_n\rangle, Y, I)$ peuvent être décrits par une FSM, obtenue en substituant chaque macro-état b par la FSM correspondant au produit asynchrone entre les FSMs données par $Y(b)$. Ainsi à chaque macro-état $b \in \mathcal{B}$, on associe sa FSM correspondante $K_b^F = \langle \Sigma_b^F, \mathcal{X}_b^F, \mathcal{X}_{o_b}^F, x_{f_b}^F, \delta_b^F\rangle = \|_{j\in J_b} G_j$. Finalement pour obtenir la FSM globale à partir de K, chaque macro-état b de $\mathcal{B}$ est remplacée par sa FSM correspondante K_b^F (les états seront notés $[b, \langle x_1, \ldots, x_{\|J_b\|}\rangle]$), la connection entre les états initiaux de K_b^F et les états de K est alors réalisée en fonction de I (resp. pour l'état final). Par la suite, on notera $\mathcal{K}^F$ cette FSM, dite FSM associée à $\mathcal{K}$. Une définition plus précise de $\mathcal{K}^F$ pour une HFSM peut se trouver dans [MAR 02].

États et Configurations. Une configuration de $\mathcal{K} = (\langle K, G_1, \ldots, G_n\rangle, Y, I)$ peut être un état atomique de K ou de la forme $[b, \langle x_{j_1}, \ldots, x_{j_{\|J_b\|}}\rangle]$ où $\{j_1, \ldots, j_{\|J_b\|}\} = J_b$, ce qui correspond intuitivement à une configuration du produit asynchrone induit par le macro-état b. $\mathcal{X}^F$ correspondra à l'ensemble des configurations atteignables de $\mathcal{K}$ (ou de manière équivalente de $\mathcal{K}^F$).

3. Notons que G_i ne possède qu'un seul état final.

4.2. *Le problème de la synthèse de superviseur pour une HFSM*

Nous considérons ici le problème du contrôle d'une HFSM $\mathcal{K}$. Notre but est de calculer un superviseur interdisant un ensemble de configurations de $\mathcal{K}$. On souhaite de plus que cet ensemble soit suffisamment général pour représenter un ensemble quelconque d'états de $\mathcal{K}^F$.

Configurations interdites. Avec les notations de la définition 5, soit une HFSM $\mathcal{K} = (\langle K, G_1, \ldots, G_n\rangle, Y, I)$ telle que $K = (\Sigma, \mathcal{X}, \mathcal{B}, \mathcal{X}_o, \mathcal{X}_f, \delta)$. Étant donné $b \in \mathcal{B}$, on note $E^b = \bigcup_{1\leq j\leq m_b} E^{b,j}$ avec $E^{b,j} = E^{b,j}_{j_1} \times \cdots \times E^{b,j}_{j_{\|J_b\|}}$ et $E^{b,j}_{j_i} \subseteq \mathcal{X}_{j_i}$ pour $j_i \in J_b$. Par simplicité, on note $[b, E^b]$ l'ensemble des configurations $[b, \langle x_{j_1}, \ldots, x_{j_{\|J_b\|}}\rangle]$ telles que $\langle x_{j_1}, \ldots, x_{j_{\|J_b\|}}\rangle \in E^b$. N'importe quel ensemble d'états de $\mathcal{K}^F$ peut être représenté par un ensemble de configurations E de $\mathcal{K}$ de la forme

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

où $E_0 \subseteq \mathcal{X}$. L'ensemble E_0 représente les configurations interdites au niveau supérieur de $\mathcal{K}$, alors que $[b, E^b]$ représente l'ensemble des configurations interdites dans le produit asynchrone de FSMs données par $Y(b)$. Comme en Section 3, on souhaite décrire le superviseur maximal assurant l'interdiction de E à partir de superviseurs calculés localement sur chaque FSM et sur la structure de haut niveau sans avoir à calculer la FSM associée[4].

Contrôle d'une structure. Dans un premier temps, nous avons besoin d'étendre les définitions d'ensemble d'états faiblement interdits de manière à prendre en compte la hiérarchie. L'idée est de ne pas interdire un macro-état dans sa globalité lorsqu'il mène à un état interdit, mais de considérer le fait qu'il a un comportement interne et donc qu'en "descendant" dans la hiérarchie, le contrôle peut s'effectuer au niveau inférieur. *A contrario*, considérons $b \in \mathcal{B}$ un macro-état de K. Par contrôle sur les structures internes à b, des configurations initiales de b peuvent devenir des configurations interdites. On peut donc être amener à empêcher l'atteignabilité de b par les événements faisant évoluer le système dans ces configurations initiales et donc à interdire partiellement le macro-état b. Dans ce but, pour $A \subseteq \delta^{-1}(b)$, on introduit $b_{|A} = (b, A)$ et $\mathcal{B}_{|\mathcal{A}} = \{b_{|A} \mid b \in \mathcal{B} \text{ et } A \subseteq \delta^{-1}(b)\}$. L'idée du contrôle sera donc d'empêcher dans K l'atteignabilité de $b_{|A}$ via des événements appartenant à l'ensemble A. Ce type particulier d'objectif sera utilisé pour empêcher certains états initiaux des structures de $Y(b)$ d'être atteignables.

Basé sur ces remarques, nous étendons les définitions d'ensemble d'états faiblement interdits de manière à prendre en compte les macro-états d'une structure.

---

4. Le comportement du système contrôlé doit être le même que celui obtenu sur $\mathcal{K}^F$.

Définition 7 *Soit* $K = \langle \Sigma, \mathcal{X}, \mathcal{B}, \mathcal{X}_o, \mathcal{X}_f, \delta \rangle$ *la structure supérieure d'une HFSM* $\mathcal{K}$ *et* $e \in \mathcal{X} \cup \mathcal{B}_{|\mathcal{A}}$.

• *Si* $e \in \mathcal{X}$, *alors*

$$\begin{array}{ll} \mathcal{I}(e) & = \{x \in \mathcal{X} | \exists s \in \Sigma_{uc}^*, \delta(s,x) = e \text{ et } \forall s' \leq s,\ \delta(s',x) \notin \mathcal{B}\}. \\ \mathcal{I}_B(e) & = \{b \in \mathcal{B} | \exists \sigma \in \Sigma_{uc}, \delta(\sigma, b) \in \mathcal{I}(e)\} \end{array}$$

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

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

Finalement, pour $E \subseteq \mathcal{X} \cup \mathcal{B}_{|\mathcal{A}}$, $\mathcal{I}(E) = \cup_{e \in E} \mathcal{I}(e)$[5], *et* $\mathcal{I}_B(E) = \cup_{e \in E} \mathcal{I}_B(e)$ •

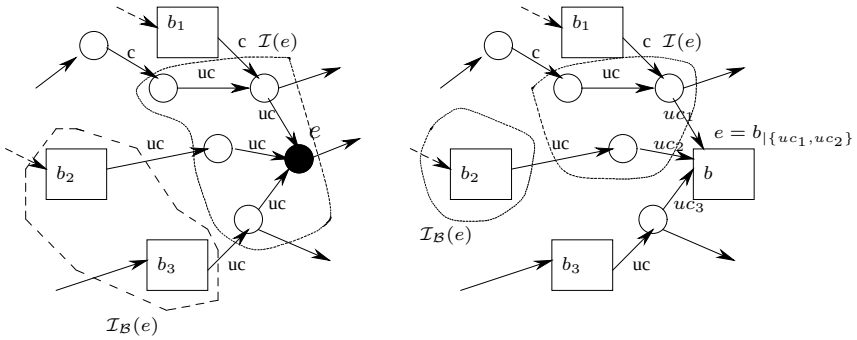


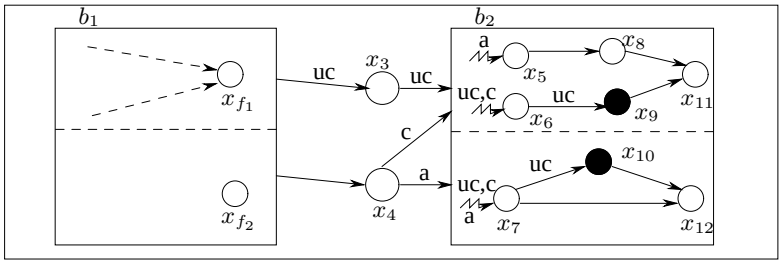
Figure 3. *Calcul de* $\mathcal{I}(e)$,$\mathcal{I}_B(e)$ *sur un exemple*

Intuitivement, si $e \in \mathcal{X}$, $\mathcal{I}(e)$ (resp. $\mathcal{I}_B(e)$) représente l'ensemble des états atomiques (resp. macro-états) de K à partir desquels e peut être atteint via une trajectoire incontrôlable qui ne traverse que des états atomiques de K. Si $e = b_{|A}$, la signification de $\mathcal{I}(e)$ et de $\mathcal{I}_B(e)$ est identique, excepté que le dernier événement de la trajectoire doit appartenir à A.

L'opérateur Φ que nous définissons maintenant sera utile pour calculer l'ensemble global des états faiblement interdits en montant/descendant dans la hiérarchie. En effet, l'interdiction d'un état initial dans un macro-état fait que l'on doit remonter au niveau supérieur pour interdire les états menant à cet état initial. Inversement, il est nécessaire d'interdire l'état final d'un macro-état menant de manière incontrôlable à un état interdit du niveau supérieur. La fonction Φ (C.f Définition 8) permet de déterminer l'ensemble des configurations de la HFSM à interdire. Avant d'introduire cette définition, l'exemple suivant illustre de manière intuitive cet aspect:

Exemple 3 *On suppose ici que l'on souhaite interdire la configuration* $[b_2, \langle x_9, x_{10} \rangle]$.

5. Notons que $\mathcal{I}$ est compatible avec la fonction décrite en Définition 4 quand $\mathcal{B} = \emptyset$.



Les calculs locaux des états faiblement interdits nous amène à considérer l'interdiction de $\{x_6, x_9\} \times \{x_7, x_{10}\}$*. On voit alors la nécessité d'empêcher le macro-état* b_2 *d'être atteignable suivant les événements qui mènent à* $[b_2, \langle x_6, x_7\rangle]$*. Compte tenu de la fonction* I*, on doit empêcher* b_2 *d'être atteint par tirage des événements* c *ou* uc*.* $b_{2|\{c,uc\}}$ *doit être interdit (Pt 2 Def. 8). De plus, compte tenu du caractère incontrôlable de* uc*, on a* $\mathcal{I}(b_{2|\{c,uc\}}) = \{x_3\}$ *et* $\mathcal{I}_B(b_{2|\{c,uc\}}) = \{b_1\}$*. On en déduit que* x_3 *et* $[b_1, \langle x_{f_1}, x_{f_2}\rangle]$ *doit être ajoutée à l'ensemble des configurations interdites (Pt 1 Def. 8). Par conséquent, étant donnée une configuration interdite, la fonction* Φ *permet de déterminer les configurations ou éléments de* $\mathcal{B}_{|\mathcal{A}}$ *devant être interdits au niveau inférieur ou supérieur pour assurer l'objectif de contrôle à partir de calculs locaux.*

Définition 8 *Soit* $e \in \mathcal{X}^F \cup \mathcal{B}_{|\mathcal{A}}$*. Pour* $b \in \mathcal{B}$*, on note* $\mathcal{X}_{ob}$ *(resp.* x_{f_b}*) l'ensemble des états initiaux (resp. l'état final) du produit asynchrone des FSMs données par* $Y(b)$*.* $\Phi(e)$ *est définie comme suit :*

1) Si $e \in \mathcal{X} \cup \mathcal{B}_{|\mathcal{A}}$*,* $\Phi(e) = \mathcal{I}(e) \cup \{[b, x_{f_b}] \mid b \in \mathcal{I}_{\mathcal{B}}(e)\}$[6]

2) Si $e = [b, \langle x_{j_1}, \ldots, x_{j_{\|J_b\|}}\rangle]$*, alors si l'on note* $\mathcal{I} = \mathcal{I}(x_{j_1}) \times \cdots \times \mathcal{I}(x_{j_{\|J_b\|}})$

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

De plus, pour $E \subseteq \mathcal{X}^F \cup \mathcal{B}_{|\mathcal{A}}$*,* $\Phi(E) = \cup_{e \in E} \Phi(e)$*.*

Soit $e \in \mathcal{X}^F \cup \mathcal{B}_{|\mathcal{A}}$. Si $e \in \mathcal{X} \cup \mathcal{B}_{|\mathcal{A}}$, alors $\Phi(e)$ correspond à l'ensemble des configurations faiblement interdites issues de e auxquelles s'ajoutent les états finals des macro-états pouvant mener de manière incontrôlable à e. Si $e = [b, \langle x_{j_1}, \ldots, x_{j_{\|J_b\|}}\rangle]$, alors $\Phi(e)$ correspond à l'ensemble des configurations faiblement interdites de e auxquelles s'ajoutent éventuellement $b_{|A} \in \mathcal{B}_{|\mathcal{A}}$ tel que A représentent l'ensemble des événements de $\delta^{-1}(b)$ menant à une configuration faiblement interdite du produit des FSMs $(G_j)_{j \in J_b}$ qui est initiale. On peut remarquer que le calcul de $\Phi(e)$ est réalisé localement.

Calcul du superviseur. Soient $\mathcal{K}$ une HFSM et $E \subseteq \mathcal{X}^F$ un ensemble de configurations de $\mathcal{K}$. On note $\mathcal{I}_H(E)$ la limite de la suite définie par $(\Phi^n(E))_{n \geq 0}$. Compte tenu de la définition de Φ et du fait que $\mathcal{K}$ possède un nombre fini de configurations, cette

6. la fonction $\mathcal{I}$ utilisée est celle décrite en Définition 7.

limite existe toujours et est obtenue en un nombre fini d'étapes. De plus les éléments retournés par Φ appartiennent à $\mathcal{X}$ ou à $\mathcal{B}_{|\mathcal{A}}$, ou sont des configurations de macro-états. Ainsi, il est possible de réorganiser l'ensemble $\mathcal{I}_H(E)$ de la manière suivante :

$$\mathcal{I}_H(E) = \left(\mathcal{X}' \cup \mathcal{B}'_{|\mathcal{A}}\right) \cup \bigcup_{b \in \mathcal{B}} [b, E^b] \qquad [6]$$

où $E^b = \bigcup_i E^{b,i}_{j_1} \times \cdots \times E^{b,i}_{j_{\|J_b\|}}$ (i.e. une union de pavés comme décrit en Section 3), $\mathcal{X}' \subseteq \mathcal{X}$ et $\mathcal{B}'_{|\mathcal{A}} \subseteq \mathcal{B}_{|\mathcal{A}}$. Le lien entre $\mathcal{I}_H(E)$ et les configurations de $\mathcal{K}^F$ menant à E de manière incontrôlable est donné par la proposition suivante.

Proposition 4 *Soit E un ensemble de configurations d'une HFSM $\mathcal{K}$ à interdire. On a $\mathcal{I}_H(E) \setminus \mathcal{B}'_{|\mathcal{A}} = \mathcal{I}(E)$ où $\mathcal{I}(E)$ est l'ensemble des configurations de $\mathcal{K}$ représentant les états faiblement interdits de E dans la FSM associée $\mathcal{K}^F$ (calculés comme en definition 4).*

Par conséquent, interdire $\mathcal{I}_H(E)$ dans $\mathcal{K}$ équivaut à interdire E dans $\mathcal{K}^F$. Le superviseur assurant l'interdiction de E dans $\mathcal{K}$ peut donc être décrit à partir des superviseurs locaux assurant l'interdiction des éléments de $\mathcal{I}_H(E)$.

1) $\forall b \in \mathcal{B}$, on calcule le superviseur $\mathcal{S}_b = (S_b, \mathcal{X}'_{0b})$ assurant l'interdiction de E^b dans $\|_{j \in J_b} G_j$, en utilisant les méthodes de la section 3 (Propositions 2 et 3).

2) Pour K, on calcule un superviseur $\mathcal{S}_K = (S_K, \mathcal{X}'_{0K})$ défini par:

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

$\mathcal{X}' \cup \mathcal{B}'_{|\mathcal{A}}$ correspondant à l'ensemble des états faiblement interdits du niveau supérieur, il est donc juste nécessaire de calculer la frontière de cet ensemble.

Proposition 5 *Avec les notations précédentes, soit $\mathcal{S}_E = (S_E, \mathcal{X}'_{oE})$ t.q.*

$$\begin{array}{l} S_E(e) = \left\{ \begin{array}{l} S_K(e) \textit{ si } e \in \mathcal{X} \\ S_K(b) \cup S_b(x_{j_1}, \ldots, x_{j_{\|J_b\|}}) \textit{ si } e = [b, \langle x_{j_1}, \ldots, x_{j_{\|J_b\|}} \rangle] \\ \emptyset \textit{ sinon} \end{array} \right. \\ \mathcal{X}_{o_E} = \mathcal{X}'_o \end{array} \qquad [7]$$

Le superviseur $\mathcal{S}_E$ assure l'interdiction de E dans $\mathcal{K}$ et est maximal. $\diamond$

Pour conclure cette section, remarquons que tout comme en Section 3, le superviseur a été synthétisé sans avoir calculé explicitement $\mathcal{K}^F$. En particulier, l'ensemble des configurations interdites a été calculé localement sur chaque $(G_i)_{1 \leq i \leq n}$ et sur la structure K.

5. Conclusion

Dans cet article, nous avons considéré le contrôle de systèmes à événements discrets modélisés par un produit asynchrone de FSMs et par des HFSMs. Des méthodes permettant de donner un superviseur maximal assurant l'interdiction d'un ensemble d'états d'un système ainsi modélisé sont décrites. Les calculs effectués sont locaux à chaque sous-système, évitant ainsi l'explosion combinatoire résultant du calcul explicite du système global. Nous travaillons actuellement sur l'extension du modèle des HFSMs (synchronisations entre les structures, possibilité de préemption, etc...) ainsi que sur des algorithmes assurant la propriété de non-blocage du système contrôlé résultant.

6. Bibliographie

[BRA 93] BRAVE Y., HEIMANN M., « Control of Discrete Event Systems Modeled as hierarchical State Machines », *IEEE Transacations on Automatic Control*, vol. 38, n° 12, 1993, p. 1803–1819.

[BRA 00] BRANDIN B., MALIK R., DIETRICH P., « Incremental System Verification and Synthesis of Minimally Restrictive Behaviours », *Proceedings of the American Control Conference*, Chicago, Illinois, Juin 2000, p. 4056-4061.

[CAS 99] CASSANDRAS C., LAFORTUNE S., *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, 1999.

[deQ 00] DEQUEIROZ M., CURY J., « Modular supervisory control of large scale discrete-event systems », *Discrete Event Systems: Analysis and Control. Proc. WODES'00*, Kluwer Academic, 2000, p. 103-110.

[deQ 02] DEQUEIROZ M., CURY J., « Synthesis and implementation of local modular supervisory control for a manufacturing cell », *Proceedings of hte 6th International Workshop on Discrete Event Systems*, October 2002, p. 377-382.

[GOH 98] GOHARI-MOGHADAM P., « A linguistic Framework for controller hierarchical DES », M.A.S.C. Thesis, Dept. of Electl. & Comptr. Engrg., University of Toronto, Avril 1998.

[HAR 85] HAREL D., PNUELI A., « On the development of Reactive Systems », *Logics and Models of Concurrent Systems*, vol. 13 de *NATO ASI Series*, New York, 1985, p. 477–498.

[LED 02] LEDUC R., « Hierarchical Interface Based Supervisory Control », PhD thesis, Dept. of Elec. & Comp. Engrg., Univ. of Toronto, 2002.

[MAR 02] MARCHAND H., GAUDIN B., « Supervisory Control Problems of Hierarchical Finite State Machines », *41th IEEE Conference on Decision and Control*, Las Vegas, USA, Decembre 2002.

[RAM 89] RAMADGE P. J., WONHAM W. M., « The Control of Discrete Event Systems », *Proceedings of the IEEE; Special issue on Dynamics of Discrete Event Systems*, vol. 77, n° 1, 1989, p. 81–98.

[WON 88] WONHAM W. M., RAMADGE P. J., « Modular Supervisory Control of Discrete Event Systems », *Mathematics of Control Signals and Systems*, vol. 1, 1988, p. 13–30.

[WON 96] WONG K., WONHAM W., « Hierarchical Control of Discrete-Event Systems », *Discrete Event Dynamic Systems*, vol. 6, 1996, p. 241-273.