



# Une approche modulaire pour le contrôle de systèmes à événements discrets concurrents

Benoit Gaudin, Hervé Marchand

► **To cite this version:**

Benoit Gaudin, Hervé Marchand. Une approche modulaire pour le contrôle de systèmes à événements discrets concurrents. 5ième Colloque Francophone sur la Modélisation des Systèmes Réactifs, Oct 2005, Autran, France. Hermès, 39, pp.79-94, 2005, Journal Européen des Systèmes Automatisés. <inria-00520028>

**HAL Id: inria-00520028**

**<https://hal.inria.fr/inria-00520028>**

Submitted on 22 Sep 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Une approche modulaire pour le contrôle de systèmes à événements discrets concurrents

**Benoit Gaudin — Hervé Marchand**

*Université de Rennes 1 & INRIA Rennes  
Équipe VerTeCs, IRISA, Campus universitaire de Beaulieu, 35042 Rennes  
{Benoit.Gaudin, Herve.Marchand}@irisa.fr*

---

*RÉSUMÉ. Dans cet article, nous nous intéressons au contrôle de systèmes à événements discrets concurrents définis par une collection de sous-systèmes interagissant les uns avec les autres. Étant donné un objectif de contrôle, le but consiste à calculer un superviseur maximal assurant cet objectif, sans construire explicitement le système à contrôler. Des approximations du système  $G$  sont dérivés à partir des sous-systèmes qui le composent, et une propriété appelée contrôlabilité partielle, devant être vérifiée par l'objectif sur ces approximations, est introduite. Assurer la contrôlabilité partielle de l'objectif sur chacune des approximations permet, sous certaines hypothèses, d'en déduire un superviseur maximal assurant l'objectif de contrôle sur  $G$ . Les calculs effectués ont une faible complexité et ne nécessitent pas de construire explicitement le système  $G$ , évitant ainsi l'explosion combinatoire inhérente aux systèmes concurrents.*

*ABSTRACT. In this paper, we are interested in the control of a particular class of Concurrent Discrete Event Systems defined by a collection of components that interact with each other. We investigate the computation of the supremal controllable language contained in the one of the specification. We do not adopt the decentralized approach. Instead, we have chosen to perform the control on some approximations of the plant derived from the behavior of each component. The behavior of these approximations is restricted so that they respect a new language property for discrete event systems called partial controllability condition that depends on the specification. It is shown that, under some assumptions, the intersection of these "controlled approximations" corresponds to the supremal controllable language contained in the specification with respect to the plant. This computation is performed without having to build the whole plant, hence avoiding the state space explosion induced by the concurrent nature of the plant.*

*MOTS-CLÉS : systèmes à événements discrets concurrents, contrôle, contrôlabilité partielle.*

*KEYWORDS: Concurrent Discrete Event Systems, control, Partial controllability.*

---

## 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/spécifications) que le système initial ne vérifiait pas. Dans cet article, nous adoptons le formalisme de (Wonham, 2003) et modélisons le système et les objectifs par des langages sur un alphabet fini. Étant donné un système  $G$  (avec  $\mathcal{L}(G)$  le langage traduisant son comportement) et une spécification  $K$  de son comportement attendu, le contrôle du système consiste en l'interdiction, par l'intermédiaire d'un superviseur  $\mathcal{S}$ , de certains événements, appelés événements contrôlables de manière à ce que le comportement du système contrôlé (noté  $\mathcal{S}/G^1$ ) soit inclus dans celui de la spécification. En utilisant la terminologie de (Wonham, 2003), le problème posé consiste donc en le calcul du plus grand langage contrôlable  $\mathcal{L}(\mathcal{S}/G)$  qui soit inclus à la fois dans le langage du système et celui de la spécification attendue. Nous nous intéressons ici au contrôle de systèmes à événements discrets concurrents définis par une collection de sous-systèmes (ou composants) interagissant les uns avec les autres. Nous nous intéressons à des systèmes dont la construction explicite est supposée impossible, du fait de l'explosion de l'espace d'états induite par la composition, rendant impossible l'utilisation des méthodologies de synthèse de contrôleurs classiques (C.f. (Wonham, 2003, Cassandras *et al.*, 1999) pour plus de détails sur ces méthodes).

Afin de répondre aux différents problèmes de complexité liés à la synthèse de contrôleurs sur un système concurrent, différentes approches prenant en compte la structure du système ont été envisagées ((Willner *et al.*, 1991), (Eyzell *et al.*, 1998), (Rohloff *et al.*, 2003), (Lee *et al.*, 2002), (deQueiroz *et al.*, 2000), (Akeson *et al.*, 2002)). Ainsi étant donné un système concurrent  $G = G_1 \parallel \dots \parallel G_n$  et  $K$  un langage modélisant la spécification, le problème est de calculer le plus grand langage contrôlable de  $K \cap \mathcal{L}(G)$  sans avoir à calculer explicitement ni  $\mathcal{L}(G)$  ni  $K \cap \mathcal{L}(G)$ . Dérivée de l'approche décentralisée, (Willner *et al.*, 1991) fournit une solution permettant de calculer localement des superviseurs  $\mathcal{S}_i$  agissant uniquement sur  $G_i$  (basé sur une notion de langage séparable<sup>2</sup>) de telle manière que le système obtenu par la mise en parallèle des systèmes contrôlés  $\mathcal{S}_i/G_i$  ainsi calculés ait un comportement qui corresponde au plus grand langage contrôlable de  $K \cap \mathcal{L}(G)$ . La même méthodologie (donnant lieu à un résultat d'existence) a été utilisée par (Rohloff *et al.*, 2003) pour le contrôle de systèmes concurrents dans lequel tous les composants possèdent une structure identique (et sous la contrainte supplémentaire que les superviseurs locaux n'observent qu'un sous-ensemble des événements locaux). Dans (deQueiroz *et al.*, 2000, Akeson *et al.*, 2002), l'idée développée par les auteurs consiste à réarranger le système par une composition parallèle dont les opérandes sont des langages sur des alphabets deux à deux disjoints, se ramenant ainsi au cas des systèmes asynchrones (product system). Cette décomposition s'effectue selon l'alphabet de l'objectif de contrôle. Une telle approche trouve notamment son intérêt lorsque les objectifs de contrôle considérés ne concernent qu'un sous-ensemble des sous-systèmes du sys-

1. dont le comportement sera noté  $\mathcal{L}(\mathcal{S}/G)$ .

2. La spécification peut se décomposer relativement aux alphabets des composants du système.

tème concurrent. Lorsque le système est obtenu par composition asynchrone de sous-systèmes et que l'objectif ne concerne qu'un seul de ces sous-systèmes, la synthèse de contrôleurs peut alors s'effectuer localement. On peut également regarder (Abdelwahed *et al.*, 2002, Jiang *et al.*, 2000, Leduc *et al.*, 2001) pour d'autres travaux reliés au contrôle de systèmes concurrents, basés sur une approche décentralisée. Dans ce papier, nous adoptons une démarche différente de celle des travaux précédemment cités. Au lieu d'avoir un superviseur par composant n'agissant que sur les événements de ce composant, nous avons choisi de calculer des superviseurs assurant l'objectif initial sur des approximations du système  $G$  qui sont dérivés à partir des sous-systèmes qui le composent. Ceci nous amène à introduire une propriété appelée *contrôlabilité partielle*, qui devra être vérifiée par l'objectif relativement à ces approximations. Assurer la contrôlabilité partielle de l'objectif de contrôle sur chacune des approximations permet, sous certaines hypothèses, d'en déduire un superviseur maximal assurant l'objectif de contrôle sur  $G$ . De plus, les calculs effectués ont une faible complexité et ne nécessitent pas de construire explicitement le système  $G$ , évitant ainsi l'explosion combinatoire inhérente aux systèmes concurrents. De plus cette technique s'applique pour des objectifs ne possédant pas nécessairement une structure similaire à celle du système.

Le reste du papier s'organise de la manière suivante : En Section 2, le modèle ainsi que les principaux concepts de la synthèse de contrôleurs sont introduits. la notion de contrôlabilité partielle est introduite en Section 3.2. Finalement, la quatrième section présente une méthode de synthèse modulaire pour des systèmes à événements discrets concurrents, ainsi que des conditions suffisantes sous lesquelles notre méthode permet de dériver un superviseur maximal permettant d'assurer de manière optimale l'objectif.

## 2. Préliminaires

**Modèle de base.** Les différents composants du système seront modélisés par des automates représentés par des quadruplets  $G = \langle \Sigma, Q, q_o, \delta \rangle$ , où  $\Sigma$  est l'alphabet fini des actions.  $\delta(\sigma, q)!$  signifie que  $\sigma$  est un événement admissible en  $q$ .  $\delta(s, q)$  représente l'état atteint par tirage de la trace  $s$  depuis l'état  $q$ . Le comportement du système est donné par le langage  $\mathcal{L}(G) \subseteq \Sigma^*$  généré par  $G$  ( $\mathcal{L}(G) = \{s \in \Sigma^* \mid \delta(s, q_o)!\}$ ). Étant données  $s, s' \in \Sigma^*$ ,  $s' \leq s$  signifie que  $s'$  est un *préfixe* de  $s$  (i.e.  $\exists t \in \Sigma^*$  t.q.  $s = s't$ ).  $\bar{L}$  correspond à la *clôture par préfixe* de  $L \subseteq \Sigma^*$  (i.e.  $\bar{L} = \{s \in \Sigma^* \mid \exists s' \in L, s \leq s'\}$ ). On peut remarquer que  $\mathcal{L}(G)$  est clos par préfixe (i.e.  $\mathcal{L}(G) = \bar{\mathcal{L}(G)}$ ). Pour  $L \subseteq \Sigma^*$  et  $\Sigma' \subseteq \Sigma$ ,  $L(s, \Sigma')$  correspond à l'ensemble des suffixes de  $L$  après la séquence  $s$  qui appartiennent à  $\Sigma'^*$ , i.e.

$$L(s, \Sigma') = \{t \in \Sigma'^* \mid st \in L\} \quad [1]$$

Soit  $\Sigma' \subseteq \Sigma$ , alors  $P_{\Sigma'} : \Sigma^* \longrightarrow \Sigma'^*$  correspond à la *projection naturelle* de  $\Sigma$  sur  $\Sigma'$  qui consiste à enlever d'une séquence donnée de  $\Sigma^*$ , tous les événements n'appartenant pas à  $\Sigma'$ . Cette définition s'étend naturellement à un langage  $L \subseteq \Sigma^*$  par  $P_{\Sigma'}(L) = \{s \in \Sigma'^* \mid \exists s' \in L, P_{\Sigma'}(s') = s\}$ . De manière duale, étant donné deux alphabets  $\Sigma'$  et  $\Sigma$ , ainsi qu'un langage  $L \subseteq \Sigma'^* \subseteq \Sigma^*$ , la *projection inverse* de  $\Sigma'$

dans  $\Sigma$ , notée  $P_{\Sigma'}^{-1}$ , est définie par  $P_{\Sigma'}^{-1}(L) = \{s \in \Sigma^* \mid P_{\Sigma'}(s) \in L\}$ . Intuitivement,  $P_{\Sigma'}^{-1}(L)$  est obtenu en insérant de toutes les manières possibles des séquences de  $(\Sigma \setminus \Sigma')^*$  dans les mots de  $L$ .

**Lemme 1** (Gaudin, 2004) Soient  $\Sigma' \subseteq \Sigma$  deux alphabets et  $L \subseteq \Sigma'^*$  un langage clos par préfixe. Soient  $s \in P_{\Sigma'}^{-1}(L)$  et  $s' \in \Sigma^*$ . Alors,  $ss' \in P_{\Sigma'}^{-1}(L) \iff sP_{\Sigma'}(s') \in P_{\Sigma'}^{-1}(L)$

**Problème de base de la synthèse de contrôleurs.** De manière à contrôler un système, son alphabet  $\Sigma$  est partitionné en deux sous-ensembles disjoints  $\Sigma_c$  et  $\Sigma_{uc}$  appelés respectivement ensemble des événements contrôlables et incontrôlables.

*Superviseur.* Le contrôle d'un système s'opère par le biais d'un superviseur dont le but est d'agir sur l'évolution du système en permettant (interdisant) l'occurrence d'événements en fonction du comportement passé du système. Formellement, un superviseur  $\mathcal{S}$  est une fonction  $\mathcal{S} : \mathcal{L}(G) \rightarrow 2^\Sigma$ , qui retourne l'ensemble des événements permis après une séquence de  $G$ . On note  $\mathcal{S}/G$  le système  $G$  contrôlé par  $\mathcal{S}$  et  $\mathcal{L}(\mathcal{S}/G)$  le langage traduisant son comportement.

*Notion de contrôlabilité.* 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.

**Définition 1** (Wonham, 2003) Soit  $G$  un système modélisé par un automate et  $K \subseteq \mathcal{L}(G)$ , clos par préfixe.  $K$  est dit contrôlable relativement à  $\Sigma_{uc}$  et  $G$  (ou  $\mathcal{L}(G)$ ) si  $K\Sigma_{uc} \cap \mathcal{L}(G) \subseteq K$  •

Par la suite,  $K^{\uparrow \mathcal{L}(G),c}$  ou  $SupC(K, \mathcal{L}(G), \Sigma_{uc})$  correspond au sous-langage maximal contrôlable de  $K$  relativement à  $\Sigma_{uc}$  et  $\mathcal{L}(G)$ <sup>3</sup>. Soit  $\mathcal{S}$  un superviseur tel que

$$\mathcal{L}(\mathcal{S}/G) = SupC(K, \mathcal{L}(G), \Sigma_{uc}) \quad [2]$$

$\mathcal{L}(\mathcal{S}/G)$  correspond au plus grand comportement de  $G$  qui vérifie la spécification  $K$  tout en respectant les contraintes de contrôlabilité. Un tel superviseur sera alors dit *maximal*. Dans certaines situations, il est également intéressant de calculer  $K^{\downarrow \mathcal{L}(G),c}$  le plus petit langage contrôlable clos par préfixe contenant  $K$  par rapport à  $\mathcal{L}(G)$  et  $\Sigma_{uc}$ . On peut montrer que (C.f. (Cassandras et al., 1999)):

$$K^{\downarrow \mathcal{L}(G),c} = K\Sigma_{uc}^* \cap \mathcal{L}(G) \quad [3]$$

**Systèmes concurrents.** Dans ce papier, nous nous intéressons au contrôle de systèmes à événements discrets concurrents définis par une collection de sous-systèmes (ou composants) interagissant les uns avec les autres par synchronisation sur des événements communs. Formellement, un système sera donné par un ensemble de  $n$  systèmes modélisés par des automates  $(G_i)_{i \leq n}$  avec  $G_i = (\Sigma_i, Q_i, q_{0i}, \delta_i)$  pour

3. Ce sous-langage maximal existe toujours. Il est en effet possible de montrer que la contrôlabilité est stable par union (C.f. (Wonham, 2003)).

$1 \leq i \leq n$ . Le système à contrôler est alors modélisé par l'automate obtenu par composition parallèle des automates  $(G_i)_{i \leq n}$ . On note  $G = (\Sigma, Q, q_0, \delta)$  cet automate avec  $G = G_1 \parallel \dots \parallel G_n$ , où l'opération  $\parallel$  est l'opération classique de composition parallèle (i.e.  $G_1 \parallel G_2$  représente le comportement concurrent de  $G_1$  et  $G_2$  avec synchronisation sur les événements communs). L'alphabet  $\Sigma$  de  $G$  est défini par  $\Sigma = \bigcup_i \Sigma_i$ . L'ensemble des événements *partagés* par les systèmes  $(G_i)_i$  (ou langages  $(\mathcal{L}(G_i))_i$ , noté  $\Sigma_s$ , est défini par l'ensemble  $\Sigma_s = \bigcup_{i \neq j} (\Sigma_i \cap \Sigma_j)$ . Finalement, le comportement de  $G = G_1 \parallel \dots \parallel G_n$  (i.e. le langage généré par  $G$ ) est donné par:

$$\mathcal{L}(G) = \mathcal{L}(G_1) \parallel \dots \parallel \mathcal{L}(G_n) = P_1^{-1}[\mathcal{L}(G_1)] \cap \dots \cap P_n^{-1}[\mathcal{L}(G_n)] \quad [4]$$

où pour  $i \in \{1, \dots, n\}$ ,  $P_i : \Sigma^* \longrightarrow \Sigma_i^*$  est la projection naturelle de  $\Sigma$  sur  $\Sigma_i$ . À noter que la même notation pour la composition de langages et d'automates est utilisée.

*Hypothèses de contrôlabilité.* Pour chaque sous-système  $G_i$ , certains événements de  $\Sigma_i$  sont incontrôlables ( $\Sigma_{i,uc}$ ) alors que les autres sont contrôlables ( $\Sigma_{i,c}$ ) et peuvent donc être empêchés sous l'action d'un superviseur. La nature des événements de  $\Sigma$  est supposée être la même quelque soit le sous-système  $G_i$ . En d'autres termes, un événement ne peut être contrôlable pour un des sous-systèmes et incontrôlable pour un autre. Formellement,

$$\forall i, j \in \{1, \dots, n\}, \Sigma_{i,uc} \cap \Sigma_{j,c} = \emptyset \quad [5]$$

Sous cette hypothèse, les événements incontrôlables et contrôlables de  $G$  sont alors respectivement donnés par  $\Sigma_{uc} = \bigcup_i \Sigma_{i,uc}$ , et  $\Sigma_c = \bigcup_i \Sigma_{i,c}$ .

On peut noter que l'hypothèse donnée par [5] est restrictive dès lors que l'on souhaite appliquer des superviseurs locaux à chacun des sous systèmes  $G_i$ . En effet, il est possible que l'occurrence de certains événements puisse être empêchée en agissant sur un sous-système alors que cela est impossible depuis un autre sous système. Toutefois, cette condition est légitime lorsque le superviseur que l'on souhaite appliquer au système agit globalement dans la mesure où si l'occurrence d'un événement peut être empêchée depuis un des sous-systèmes, elle peut être empêchée de manière globale, et l'événement correspondant peut alors être supposé contrôlable dans chacun des sous-systèmes.

### 3. Contrôle de systèmes à événements discrets structurés

Soient  $G = G_1 \parallel \dots \parallel G_n$  un système concurrent et  $K \subseteq \Sigma^*$  un langage clos par préfixe modélisant la spécification attendue. Le problème qui nous intéresse est de calculer un superviseur qui réduise le comportement du système à un sous-comportement maximal vérifiant la spécification. Plus formellement, le problème est donc de calculer le plus grand langage contrôlable  $\mathcal{L}(S/G) = (K \cap \mathcal{L}(G))^{\uparrow c}$  inclus dans  $K \cap \mathcal{L}(G)$ . Toutefois, du fait du caractère concurrent des systèmes considérés, la construction explicite du système peut s'avérer problématique voir impossible, du fait de l'explosion de l'espace d'états induite par la composition, rendant impossible l'utilisation des méthodologies de synthèse de contrôleurs classiques pour le calcul du superviseur  $S$  qui est dérivé de  $K$  et de  $\mathcal{L}(G)$ . Il apparaît donc important de trouver des algorithmes qui

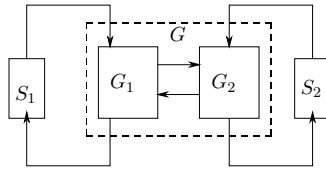
tirent avantage de la structure du système et qui permettent de calculer  $(K \cap \mathcal{L}(G))^{\uparrow c}$  sans avoir à calculer explicitement ni  $\mathcal{L}(G)$  ni  $K \cap \mathcal{L}(G)$ .

### 3.1. Différentes approches possibles

Avant de présenter notre méthodologie basée sur une approche modulaire en Section 3.1.2, nous rappelons brièvement les travaux de (Willner *et al.*, 1991) qui adoptent une approche décentralisée<sup>4</sup>.

#### 3.1.1. L'approche décentralisée

De par la nature concurrente des systèmes considérés, il est relativement intuitif d'utiliser une approche décentralisée pour la résolution de problèmes de contrôle sur les systèmes concurrents. Ainsi, dans (Willner *et al.*, 1991), les auteurs s'intéressent au contrôle d'un système concurrent  $G_1 \parallel \dots \parallel G_n$ . Étant donné un langage  $K$  modélisant le comportement attendu du système, il fournissent une solution permettant de calculer localement des superviseurs  $\mathcal{S}_i$  agissant uniquement sur  $G_i$  (basée sur une notion de langage séparable) de manière à ce que le système obtenu par la mise en parallèle des systèmes contrôlés ainsi calculés  $\mathcal{S}_i/G_i$  ait un comportement qui corresponde au plus grand langage contrôlable de  $K \cap \mathcal{L}(G)$  par rapport à  $\mathcal{L}(G)$ .



**Figure 1.** Approche dérivée de la synthèse décentralisée

**Définition 2**  $\mathcal{L} \subseteq \Sigma^*$  est séparable relativement à  $\{\Sigma_i\}_{i \leq n}$  avec  $\cup_{i \leq n} \Sigma_i = \Sigma$ , si il existe un ensemble de langages  $\{\mathcal{L}_i\}_{i \leq n}$ , t.q.  $\mathcal{L}_i \subseteq \Sigma_i^*$  et  $\mathcal{L} = \mathcal{L}_1 \parallel \dots \parallel \mathcal{L}_n$ .

La notion de langage séparable s'avère être la propriété principale sur laquelle se basent les auteurs de (Willner *et al.*, 1991) pour le calcul d'une solution maximale.

**Théorème 1** (Willner *et al.*, 1991) Soit  $G = G_1 \parallel \dots \parallel G_n$  t.q.  $\mathcal{L}(G_i) \subseteq \Sigma_i^*$ . Soit  $K \subseteq \Sigma^*$  la spécification attendue. Si  $\Sigma_s \subseteq \Sigma_c$  et  $K$  est séparable relativement à  $\{\Sigma_i\}_{i \leq n}$ , alors

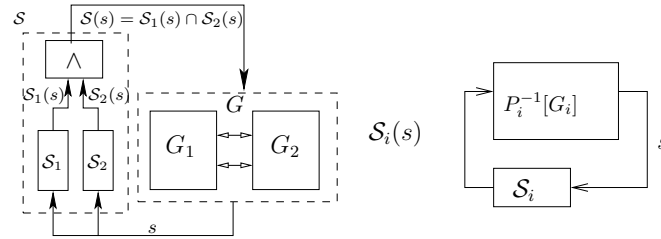
$$\|_{i \leq n} \text{SupC}(P_i(K) \cap \mathcal{L}(G_i), \mathcal{L}(G_i), \Sigma_{i,uc}) = \text{SupC}(K \cap \mathcal{L}(G), \mathcal{L}(G), \Sigma_{uc}) \quad \diamond$$

Ainsi, étant donné un système concurrent  $G$  et une spécification séparable  $K$ , le théorème 1 montre qu'il existe un ensemble de superviseurs  $\mathcal{S}_i$  agissant sur  $G_i$ , tel que  $\mathcal{L}(\mathcal{S}_i/G_i) = \text{SupC}(P_i(K) \cap \mathcal{L}(G_i), \mathcal{L}(G_i), \Sigma_{i,uc})$  et  $\|_{i \leq n} \mathcal{L}(\mathcal{S}_i/G_i) = (K \cap \mathcal{L}(G))^{\uparrow c}$ . On peut noter que si  $K$  est donné directement sous la forme  $K_1 \parallel \dots \parallel K_n$  avec  $K_i \subseteq \Sigma_i^*$ , alors le théorème 1 s'applique directement. L'architecture de contrôle est alors donnée par la figure 1.

4. Les travaux de (Eyzell *et al.*, 1998), (Rohloff *et al.*, 2003), (Lee *et al.*, 2002), (deQueiroz *et al.*, 2000), (Akesson *et al.*, 2002) adoptent également une telle approche et ne seront pas rappelés dans ce papier

### 3.1.2. Une approche centralisée modulaire

À l'inverse des approches décentralisées classiquement utilisées pour le contrôle systèmes concurrents, notre approche est différente et s'apparente plus à une approche modulaire. Plutôt que de dériver différents objectifs locaux suivant chacun des sous-systèmes  $(G_i)_{i \leq n}$ , l'idée consiste à sur-approximer le système global  $G$  à partir de chacun des sous-systèmes  $(G_i)_{i \leq n}$ , et à considérer l'objectif de contrôle initial sur chacune de ces sur-approximations. L'idée est alors de calculer à partir de ces sur-approximations  $n$  superviseurs  $(S_i)_{i \leq n}$ , où chaque superviseur est en mesure d'observer le système global. Par la suite, ces superviseurs se mettent d'accord pour savoir quels sont les événements qui sont permis après une trace donnée.



**Figure 2.** Architecture de contrôle modulaire et calcul des  $S_i$  à partir de  $P_i^{-1}(G_i)$

D'un point de vue calcul,  $G$  peut être décrit par  $G = \parallel_{i \leq n} G_i^{-1}$ , où  $G_i^{-1}$  est l'automate minimal tel que  $\mathcal{L}(G_i^{-1}) = P_i^{-1}(\mathcal{L}(G_i))$ . En fait, chaque  $G_i^{-1}$  peut être vu comme une approximation du système  $G$  à contrôler. Par rapport à (Willner *et al.*, 1991), au lieu de contrôler chaque composant  $G_i$  (i.e.  $\mathcal{L}(G_i)$ ) de manière à assurer  $P_i(K)$ , nous avons choisi de contrôler les  $G_i^{-1}$  et d'assurer  $K$  sur ceux-ci. Toutefois, il n'est pas suffisant de calculer un superviseur  $S_i$  agissant sur  $G_i^{-1}$  qui restreigne le comportement de  $\mathcal{L}(G_i^{-1})$  au plus grand langage contrôlable contenue dans  $K \cap \mathcal{L}(G_i^{-1})$  par rapport à  $\mathcal{L}(G_i^{-1})$  et  $\Sigma_{i,uc}$  (resp.  $\Sigma_{uc}$ ), et de les faire coopérer de manière modulaire. Le résultat ne serait pas contrôlable (resp. pas maximal). Ainsi l'idée est de raffiner la notion de contrôlabilité de manière à prendre en compte le fait qu'il existe des événements incontrôlables locaux à un composant. La propriété que l'on va chercher à assurer sur  $G_i^{-1}$  relativement à  $K$  est appelée *condition de contrôlabilité partielle* et est définie en Section 3.2.

### 3.2. Contrôlabilité partielle

Dans cette section, nous introduisons un nouveau concept de contrôlabilité, nommé *contrôlabilité partielle*, qui sera utilisé pour le calcul modulaire de superviseurs agissant sur des systèmes concurrents.

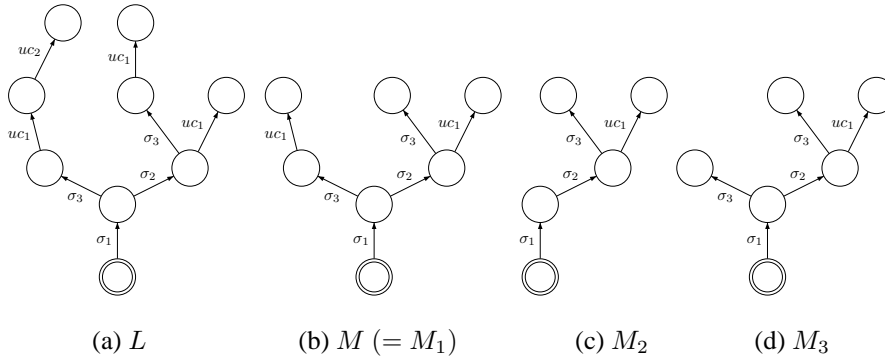
**Définition 3**  $\emptyset \subset M \subseteq L \subseteq \Sigma^*$ ,  $M$ ,  $L$  clos par préfixe,  $M' \subseteq M$ ,  $\Sigma'_{uc} \subseteq \Sigma_{uc} \subseteq \Sigma$ .  $M'$  est partiellement contrôlable par rapport à  $\Sigma'_{uc}$ ,  $\Sigma_{uc}$ ,  $M$  et  $L$  si

- (i)  $M'$  est contrôlable par rapport à  $\Sigma'_{uc}$  et  $L$ .
- (ii)  $M'$  est contrôlable par rapport à  $\Sigma_{uc}$  et  $M$ .



Intuitivement,  $L$  sera vue comme une approximation du système relativement à un des composants du système et  $M$  comme l'objectif de contrôle (C.f. Section 3.3.1 et Théorème 2). Ainsi étant donné un sous-comportement  $M'$  de  $M$ , l'idée est que l'on peut se permettre de sortir de  $M'$  par des événements incontrôlables qui ne sont pas locaux sachant que ce sera les superviseurs contrôlant les autres approximations qui vont se charger d'interdire ces événements. En revanche, l'exigence est plus forte concernant la contrôlabilité de  $M'$  par rapport à l'objectif  $M$ , dans la mesure où  $M$  n'est pas une approximation mais l'objectif exact qui doit être assuré. C'est pourquoi on souhaite que cette contrôlabilité soit vérifiée pour l'ensemble  $\Sigma_{uc}$  des événements incontrôlables. Par conséquent, la contrôlabilité partielle utilise l'objectif de contrôle  $M$  et un sous-ensemble des événements incontrôlables  $\Sigma'_{uc}$  pour affaiblir la notion de contrôlabilité d'un langage  $M'$  par rapport à  $L$ .

**Exemple 1** *Considérons les langages  $L$ ,  $M (= M_1)$ ,  $M_2$  et  $M_3$  décrit par la Figure 3. Supposons que  $\Sigma_{uc} = \{uc_1, uc_2\}$  et  $\Sigma'_{uc} = \{uc_2\}$ .*



**Figure 3.** *Comportements de  $L$ ,  $M$ ,  $M_2$ ,  $M_3$ .*

–  $M_1 = M$  n'est pas partiellement contrôlable par rapport à  $\Sigma'_{uc}$ ,  $\Sigma_{uc}$ ,  $M$  et  $L$ , puisque  $M$  n'est pas contrôlable par rapport à  $\Sigma'_{uc}$  et  $L$ . En effet,  $\sigma_1\sigma_3uc_1 \in M_1$  et  $\sigma_1\sigma_3uc_1uc_2 \in L$  avec  $uc_2 \in \Sigma'_{uc}$ , mais  $\sigma_1\sigma_3uc_1uc_2 \notin M_1$ . Cependant,  $M_1 = M$  est contrôlable par rapport à  $\Sigma_{uc}$  et  $M$ .

– A contrario,  $M_2$  est partiellement contrôlable par rapport à  $\Sigma'_{uc}$ ,  $\Sigma_{uc}$ ,  $M$  et  $L$ . En effet, d'une part  $M_2$  est contrôlable par rapport à  $\Sigma'_{uc}$ , et  $L$  (notons que puisque  $uc_1 \notin \Sigma'_{uc}$ , le fait que  $\sigma_1\sigma_2\sigma_3uc_1$  n'appartienne pas à  $M_2$  ne pose pas de problème de contrôlabilité). D'autre part,  $M_2$  est contrôlable par rapport à  $\Sigma_{uc}$ , et  $M$ . Cependant,  $M_2$  n'est pas contrôlable par rapport à  $\Sigma_{uc}$ , et  $L$ .

– Finalement,  $M_3$  est contrôlable par rapport à  $\Sigma'_{uc}$ , et  $L$ . Mais  $M_3$  n'est pas partiellement contrôlable par rapport à  $\Sigma'_{uc}$ ,  $\Sigma_{uc}$ ,  $M$  et  $L$ . En effet,  $M_3$  n'est pas contrôlable par rapport à  $\Sigma_{uc}$ , et  $M$  puisque  $\sigma_1\sigma_3 \in M_3$ ,  $\sigma_1\sigma_3uc_1 \in M$  où  $uc_1 \in \Sigma_{uc}$ , mais  $\sigma_1\sigma_3uc_1 \notin M_3$ .  $\diamond$

En général,  $M$  n'est pas partiellement contrôlable par rapport à  $\Sigma'_{uc}$ ,  $\Sigma_{uc}$ ,  $M$  et  $L$ . Toutefois, on peut montrer qu'il existe un plus grand sous-langage partiellement contrôlable de  $M$  qui a cette propriété.

**Proposition 1** (Gaudin et al., 2004b)  $\emptyset \subset M \subseteq L \subseteq \Sigma^*$ ,  $M$  et  $L$  clos par préfixe,  $\Sigma'_{uc} \subseteq \Sigma_{uc} \subseteq \Sigma$ . Il existe un unique langage maximal inclus dans  $M$  qui soit partiellement contrôlable par rapport  $\Sigma'_{uc}$ ,  $\Sigma_{uc}$ ,  $M$  et  $L$ . De plus, ce langage est

$$M^{\uparrow pc} = \overline{M^{\uparrow pc}} = \text{SupC}(\text{SupC}(M, \Sigma'_{uc}, L), \Sigma_{uc}, M) \quad [6]$$

**Exemple 2** En reprenant l'exemple 1, le plus grand sous-langage de  $M$  qui soit partiellement contrôlable par rapport à  $\Sigma'_{uc}$ ,  $\Sigma_{uc}$ , et  $L$  est le langage  $M_2$ .  $\diamond$

La proposition 1 fournit non seulement un résultat d'existence du plus grand sous-langage de  $M$  par rapport à  $\Sigma'_{uc}$ ,  $\Sigma_{uc}$ , et  $L$ , mais aussi une expression de celui-ci. Ce langage sera noté  $M^{\uparrow pc}$  lorsqu'il n'y aura pas d'ambiguïté.  $M^{\uparrow pc}$  est calculé en s'appuyant sur l'algorithme de calcul associé à l'opérateur  $\text{SupC}(\bullet, \bullet, \bullet)$ . En fait, il est possible de montrer que la complexité de ce calcul vaut  $\mathcal{O}(|\Sigma|N_M.N_L)$ , où  $N_M$  et  $N_L$  sont respectivement le nombre d'états des automates codant les langages  $M$  et  $L$ .

### 3.3. Résolution du problème de contrôle pour des systèmes concurrents

Étant donné un système concurrent  $G = G_1 \parallel \dots \parallel G_n$  et une spécification  $K$ , nous cherchons ici à calculer le plus grand sous-langage contrôlable de  $K \cap \mathcal{L}(G)$  relativement à  $\mathcal{L}(G)$  et  $\Sigma_{uc}$ , sans avoir à calculer/construire explicitement  $G$ .

#### 3.3.1. Calcul modulaire d'un sous-langage contrôlable de $K$ relativement à $\mathcal{L}(G)$

Basé sur le concept de la contrôlabilité partielle appliqué sur  $K$  et sur les approximations du système  $P_i^{-1}(\mathcal{L}(G_i))$  dérivées des composants du système, le théorème suivant fournit une méthode modulaire permettant de calculer un sous-langage de  $K$  contrôlable relativement à  $\mathcal{L}(G)$  et  $\Sigma_{uc}$ .

**Théorème 2** (Gaudin et al., 2004a) Soit  $G = G_1 \parallel \dots \parallel G_n$ , avec  $G_i$  ayant comme alphabet  $\Sigma_i = \Sigma_{i,uc} \cup \Sigma_{i,c}$ . Soit  $K \subseteq \Sigma^*$  clos par préfixe. Pour  $i \leq n$ , on note

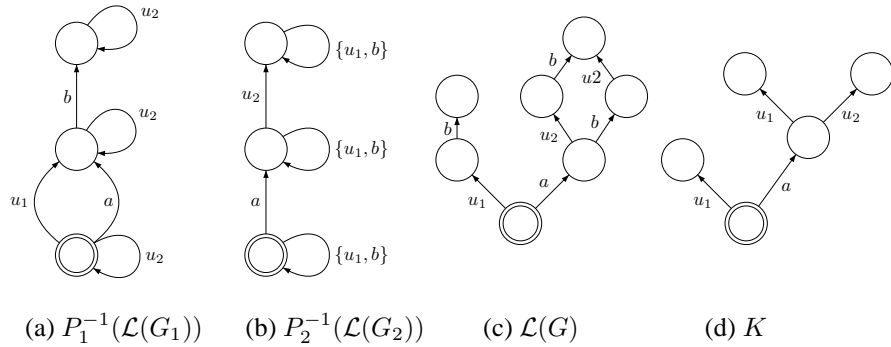
- $K_i = K \cap P_i^{-1}(\mathcal{L}(G_i))$ , et
- $K_i^{\uparrow pc}$  le sous-langage maximal de  $K_i$  partiellement contrôlable relativement à  $\Sigma_{i,uc}$ ,  $\Sigma_{uc}$ ,  $K_i$  et  $P_i^{-1}(\mathcal{L}(G_i))$ .

Alors,  $K_1^{\uparrow pc} \cap \dots \cap K_n^{\uparrow pc} \subseteq K \cap \mathcal{L}(G)$  est contrôlable relativement à  $\Sigma_{uc}$  et  $\mathcal{L}(G)$ .

Le théorème 2 nous fournit une méthode modulaire permettant de calculer un sous-langage de  $K$  contrôlable relativement à  $\mathcal{L}(G)$  et  $\Sigma_{uc}$ . D'un point de vue complexité, d'après ce qui a été vu dans la section 3.2, pour  $i \leq n$  le calcul de  $K_i^{\uparrow pc}$  a une complexité dans le pire cas égale à  $\mathcal{O}(|\Sigma|.N_K.N)$  où  $N_K$  et  $N$  représentent respectivement le nombre d'états des automates modélisant les langages  $K$  et  $\mathcal{L}(G_i)$ . Donc la complexité du calcul de l'ensemble des langages  $(K_i^{\uparrow pc})_{1 \leq i \leq n}$  vaut  $\mathcal{O}(n.|\Sigma|.N_K.N)$ . Cette complexité doit être comparée au calcul de  $(K \cap \mathcal{L}(G))^{\uparrow c}$  effectué de manière classique, i.e  $\mathcal{O}(|\Sigma|.N_K.(N^n))$ .

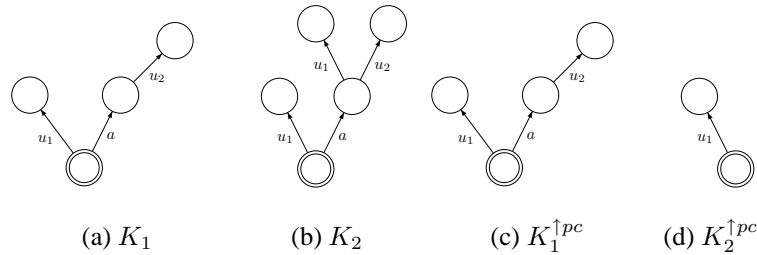
**Exemple 3** Soit  $\mathcal{L}(G) = \mathcal{L}(G_1) \parallel \mathcal{L}(G_2)$ , où  $\mathcal{L}(G_1) = \overline{\{(a + u_1)b\}}$  et  $\mathcal{L}(G_2) = \{a, au_2\}$ . On considère  $\Sigma_1 = \{a, u_1, b\}$ ,  $\Sigma_2 = \{a, u_2\}$ . Les Figures 4(a) et 4(b)

représentent les automates générant  $P_1^{-1}(\mathcal{L}(G_1))$  et  $P_2^{-1}(\mathcal{L}(G_2))$ , et la Figure 4(c) est un automate modélisant  $G$  (notons que les automates  $G_1$  et  $G_2$  sont facilement obtenus à partir des Figures 4(a) et 4(b) en enlevant les boucles d'événements). Finalement, dans cet exemple,  $\Sigma_s$  est réduit au singleton  $\{a\}$ . Supposons que l'objectif de contrôle soit donné par le langage  $K$  décrit en Figure 4(d).



**Figure 4.** Comportements du système  $G$  et de la objectif global  $K$ .

Afin d'appliquer le théorème 2, on calcule les langages  $K_i = K \cap P_i^{-1}(\mathcal{L}(G_i))$ ,  $i = 1, 2$ . Les automates générés  $K_1$  et  $K_2$  sont représentés par les Figures 5(a,b).



**Figure 5.** spécifications partielles et langages partiellement contrôlables associés

On calcule à présent  $K_1^{\uparrow pc}$  (resp  $K_2^{\uparrow pc}$ ), le plus grand langage de  $K_1$  (resp.  $K_2$ ) qui est partiellement contrôlable par rapport à  $P_1^{-1}(\mathcal{L}(G_1))$ ,  $\Sigma_{uc} = \{u_1, u_2\}$  et  $\Sigma_{1,uc} = \{u_1\}$  (resp  $\Sigma_{2,uc} = \{u_2\}$  et  $P_2^{-1}(\mathcal{L}(G_2))$ ). L'intersection de ces deux langages donne le langage  $K_1^{\uparrow pc} \cap K_2^{\uparrow pc}$  (Figure 5(d)), qui est contrôlable par rapport à  $\mathcal{L}(G)$  et  $\Sigma_{uc}$ .

◇

### 3.3.2. Architecture du superviseur agissant sur $G$

Dans cette section, nous décrivons maintenant comment un superviseur peut être extrait à partir des langages précédemment calculés et comment un tel superviseur peut agir sur  $G$  de manière à restreindre son comportement à celui de  $K$ . Avec les notations du théorème 2,  $\bigcap_{i \leq n} K_i^{\uparrow pc}$  est contrôlable relativement à  $\Sigma_{uc}$  et  $\mathcal{L}(G)$ . Toutefois, il n'est pas intéressant de calculer explicitement l'intersection de ces langages et d'en

dériver un superviseur. En s'inspirant du concept de modularité décrit dans (Wonham, 2003), le superviseur  $\mathcal{S}$  va prendre ses décisions en fonction de l'historique du système et de "superviseurs"  $\mathcal{S}_i$  dérivés de  $K_i^{\uparrow pc}$ . L'architecture du contrôle est donnée par la figure 2. Pour chaque  $K_i^{\uparrow pc}$ , nous dérivons un "superviseur"  $\mathcal{S}_i$ , qui après une trace de  $G$  (qui est également une trace de  $P_i^{-1}(\mathcal{L}(G_i))$ ), délivre l'ensemble des événements qui étendent  $s$  en une trace de  $K_i^{\uparrow pc}$  (chacun de ces superviseurs assure sur  $P_i^{-1}(\mathcal{L}(G_i))$  la propriété de contrôlabilité partielle relativement à  $K$  et  $P_i^{-1}(\mathcal{L}(G_i))$ ). Par la suite, l'ensemble des événements admissibles sous l'action du superviseur en  $G$  après la trace  $s$  est donné par  $\mathcal{S}(s) = \mathcal{S}_1(s) \cap \dots \cap \mathcal{S}_n(s)$ .

### 3.3.3. Calcul de la solution maximale $(K \cap \mathcal{L}(G))^{\uparrow c}$

Selon le théorème 2, nous savons que  $\bigcap_{i \leq n} K_i^{\uparrow pc}$  est inclus dans  $(K \cap \mathcal{L}(G))^{\uparrow c}$  et est contrôlable relativement à  $\Sigma_{uc}$  et  $\mathcal{L}(G)$ . Toutefois, en général, l'égalité n'est pas vérifiée (Voir l'exemple 3, pour lequel le sous-langage contrôlable maximal de  $K \cap \mathcal{L}(G)$  est donné par  $(K \cap \mathcal{L}(G))^{\uparrow c} = \{a, u_1, au_2\}$ ). Dans cette section, nous présentons des conditions sous lesquelles le théorème 2 fournit la solution maximale.

Dans un premier temps, nous introduisons le lemme suivant, qui nous indique que lorsque les événements partagés  $\Sigma_s$  sont contrôlables, alors  $(K \cap \mathcal{L}(G))^{\uparrow c}$  vérifie une partie des conditions de la contrôlabilité partielle.

**Lemma 1** (Gaudin et al., 2004a) Soient  $G = G_1 \parallel \dots \parallel G_n$  et  $K \subseteq \Sigma^*$ , clos par préfixe. Si  $\Sigma_s \subseteq \Sigma_c$ , alors  $(K \cap \mathcal{L}(G))^{\uparrow c}$  est contrôlable relativement à  $\Sigma_{i,uc}$ ,  $P_i^{-1}(\mathcal{L}(G_i))$ .

L'hypothèse " $\Sigma_s \subseteq \Sigma_c$ " est souvent rencontrée dans les problèmes de synthèse de superviseurs sur des systèmes concurrents. Elle est ainsi admise dans les travaux de (Willner et al., 1991) et (Abdelwahed et al., 2002). De plus, on peut noter que cette condition est toujours vérifiée lorsque les sous-systèmes ne partagent aucun événement (i.e  $\Sigma_s = \emptyset$ ) comme dans les travaux de (Minhas, 2002).

**Théorème 3** Si  $\Sigma_s \subseteq \Sigma_c$  et  $K \subseteq \mathcal{L}(G)$ , alors avec les notations du théorème 2,  $\bigcap_{i \leq n} K_i^{\uparrow pc} = K^{\uparrow \mathcal{L}(G),c} = SupC(K, \mathcal{L}(G), \Sigma_{uc})$ .

**Preuve :** Tout d'abord, on a  $\bigcap_{1 \leq i \leq n} K_i^{\uparrow pc} \subseteq \bigcap_{1 \leq i \leq n} K_i = K \cap \mathcal{L}(G)$ . De plus, d'après le théorème 2,  $\bigcap_{i \leq n} K_i^{\uparrow pc}$  est contrôlable par rapport à  $\Sigma_{uc}$  et  $\mathcal{L}(G)$ . Par maximalité de  $(K \cap \mathcal{L}(G))^{\uparrow c}$ ,  $\bigcap_i K_i^{\uparrow pc} \subseteq (K \cap \mathcal{L}(G))^{\uparrow c}$ . Afin de montrer l'inclusion inverse, il suffit de montrer que  $\forall i \leq n$ ,  $(K \cap \mathcal{L}(G))^{\uparrow c} \subseteq K_i^{\uparrow pc}$ . Or pour  $i \leq n$

$$(K \cap \mathcal{L}(G))^{\uparrow c} \subseteq K \cap \mathcal{L}(G) \subseteq K \cap P_i^{-1}(\mathcal{L}(G_i)) = K_i$$

Donc par maximalité de  $K_i^{\uparrow pc}$  (i.e  $K_i^{\uparrow pc}$  est le plus grand sous langage de  $K_i$  qui est partiellement contrôlable par rapport à  $\Sigma_{i,uc}$ ,  $\Sigma_{uc}$ ,  $K_i$ ,  $P_i^{-1}(\mathcal{L}(G_i))$ ), il suffit de montrer que  $(K \cap \mathcal{L}(G))^{\uparrow c}$  est partiellement contrôlable par rapport à  $\Sigma_{i,uc}$ ,  $\Sigma_{uc}$ ,  $K_i$ ,  $P_i^{-1}(\mathcal{L}(G_i))$ . Pour cela, on va montrer que  $(K \cap \mathcal{L}(G))^{\uparrow c}$  vérifie les deux points de la

définition 3. Le point (i) nous est donné par le lemme 1. Concernant le point (ii), par définition,  $(K \cap \mathcal{L}(G))^{\uparrow c}$  est contrôlable par rapport à  $\mathcal{L}(G)$  et  $\Sigma_{uc}$ . De plus, puisque  $K \cap \mathcal{L}(G)^{\uparrow c} \subseteq K$  et  $K \subseteq \mathcal{L}(G)$ , on déduit que  $(K \cap \mathcal{L}(G))^{\uparrow c}$  est contrôlable par rapport à  $K$  et  $\Sigma_{uc}$ . Or  $K = K \cap P_i^{-1}(\mathcal{L}(G_i)) = K_i$  puisque  $K$  est préfixe-clos, inclus dans  $\mathcal{L}(G)$ , et que  $\mathcal{L}(G) \subseteq P_i^{-1}(\mathcal{L}(G_i))$  et on obtient que  $(K \cap \mathcal{L}(G))^{\uparrow c}$  est contrôlable par rapport à  $K_i$  et  $\Sigma_{uc}$ . D'où le résultat.  $\diamond$

Le théorème 3 donne donc une condition sur le langage représentant la spécification, qui permet de résoudre notre problème de contrôle en n'effectuant que des calculs locaux sur les approximations de chaque composant avec une complexité de  $\mathcal{O}(n \cdot |\Sigma| \cdot N_K \cdot N)$  (C.f. Section 3.3.1). Il est toutefois nécessaire de vérifier au préalable l'inclusion de la spécification dans le comportement du système. Toutefois, de manière à vérifier  $K \subseteq \mathcal{L}(G)$ , on peut remarquer qu'il est suffisant de vérifier que  $\forall 1 \leq i \leq n, K \subseteq P_i^{-1}(\mathcal{L}(G_i))$ . Il n'est donc pas nécessaire de calculer explicitement  $\mathcal{L}(G)$ .

Dans certaines situations, modéliser la spécification par un langage qui doit être inclus dans celui du système peut mener à un langage qui est trop grand pour être efficacement représenté. De plus, cette inclusion de langages peut s'avérer problématique à assurer dans la mesure où le comportement du système lui-même n'est pas connu (ou du moins pas calculé). C'est dans cette optique que nous nous sommes intéressés à une autre condition de maximalité. Cette condition, appelée *cohérence locale* est définie à partir de la notion de *cohérence*, dont la définition est donnée maintenant.

**Définition 4** Soient  $\Sigma' \subseteq \Sigma$  deux alphabets et  $M \subseteq \Sigma^*$  un langage préfixe-clos. On note  $P_{\Sigma'} : \Sigma^* \rightarrow \Sigma'^*$  la projection naturelle sur  $\Sigma'$ . Soient  $\Sigma_{uc} \subseteq \Sigma$  et  $\Sigma'_{uc} = \Sigma_{uc} \cap \Sigma'$ .  $M$  est cohérent par rapport à  $P_{\Sigma'}, \Sigma_{uc}$  si

$$\forall s \in M, \forall s' \in M(s, \Sigma_{uc}), \forall \sigma \in \Sigma'_{uc}, \quad P_{\Sigma'}(s')\sigma \in M(s, \Sigma'_{uc}) \Rightarrow s'\sigma \in M(s, \Sigma_{uc}) \quad [7]$$

Cette définition permet de traduire une certaine indépendance (et donc un entrelacement) entre les événements incontrôlables locaux ( $\Sigma'_{uc}$ ) et globaux. La condition induit en particulier que si après une séquence de  $s$  dans  $M$ , il existe un événement incontrôlable local qui est tirable, alors celui-ci sera tirable tant que le système  $M$  ne tirera que des événements incontrôlables globaux appartenant à  $\Sigma_{uc} \setminus \Sigma'_{uc}$ .



**Définition 5** Soit  $G = G_1 \parallel \dots \parallel G_n$  avec  $\Sigma = \bigcup_i \Sigma_i$ ,  $\Sigma_{uc} = \bigcup_i \Sigma_{i,uc}$ . Soit  $K \subseteq \Sigma^*$ . Pour  $i \leq n$ , on note  $K_i = K \cap P_i^{-1}(\mathcal{L}(G_i))$ . Alors,  $K$  est localement cohérent par rapport à  $\Sigma_{uc}$  et  $G$  si

$$\forall i \in \{1, \dots, n\}, K_i \text{ est cohérent par rapport à } P_i \text{ et } \Sigma_{uc} \quad [8]$$

Intuitivement, si  $K$  est *localement cohérent* par rapport à  $\Sigma_{uc}$  et  $G$ , alors cela signifie que les entrelacements possibles entre événements incontrôlables locaux/globaux relativement à chaque approximation sont pris en compte dans  $K$ . Ainsi, la condition [8] constitue un relâchement des contraintes d'indépendance imposées sur un langage de Mazurkiewicz avec indépendance entre les événements incontrôlables de chaque sous-système. Dans notre cas, cette indépendance n'est exigée que pour des événements prolongeant des séquences dont on ignore si elles appartiennent ou non au système global. D'un point de vue vérification, tester la cohérence locale de  $K$  (généralisé par un automate possédant  $N_K$  états) par rapport à  $\Sigma_{uc}$  et  $G$  a une complexité en  $\mathcal{O}(n \cdot |\Sigma| \cdot (N_K \cdot N)^3)$  (Gaudin, 2004).

**Théorème 4** Si  $\Sigma_s \subseteq \Sigma_c$  et si  $K$  est localement cohérent par rapport à  $\Sigma_{uc}$  et  $\mathcal{L}(G)$ , alors, avec les notations du théorème 2,

$$\bigcap_{1 \leq i \leq n} K_i^{\uparrow pc} = (K \cap \mathcal{L}(G))^{\uparrow c}$$

**Preuve :** Soit  $K \subseteq \Sigma^*$  un langage localement cohérent par rapport à  $\Sigma_{uc}$  et  $\mathcal{L}(G)$ . On veut montrer que  $\bigcap_{i \leq n} K_i^{\uparrow pc} = (K \cap \mathcal{L}(G))^{\uparrow c}$ . Or, d'après le théorème 2, on sait que  $\bigcap_{i \leq n} K_i^{\uparrow pc} \subseteq (K \cap \mathcal{L}(G))^{\uparrow c}$  et que  $\bigcap_{i \leq n} K_i^{\uparrow pc}$  est contrôlable relativement à  $\mathcal{L}(G)$  et  $\Sigma_{uc}$ . Il suffit donc de montrer que  $\forall i \leq n$ ,  $(K \cap \mathcal{L}(G))^{\uparrow c} \subseteq K_i^{\uparrow pc}$ . On considère ainsi  $i \in \{1, \dots, n\}$  et on cherche à présent à montrer que  $(K \cap \mathcal{L}(G))^{\uparrow c} \subseteq K_i^{\uparrow pc}$ . On considère à présent  $((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c}$  et on rappelle que  $(K \cap \mathcal{L}(G))^{\uparrow c} \subseteq ((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c^5}$ . On cherche alors à montrer que  $((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c}$  est partiellement contrôlable par rapport à  $\Sigma_{i,uc}$ ,  $\Sigma_{uc}$ ,  $K_i$  et  $P_i^{-1}(\mathcal{L}(G_i))$ . En effet, dans ce cas,  $((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c} \subseteq K_i^{\uparrow pc}$ . De plus, comme  $(K \cap \mathcal{L}(G))^{\uparrow c} \subseteq ((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c}$  par définition, on obtiendrait alors que  $(K \cap \mathcal{L}(G))^{\uparrow c} \subseteq K_i^{\uparrow pc}$  et la preuve serait finie.

Montrons maintenant que  $((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c}$  est partiellement contrôlable par rapport à  $\Sigma_{i,uc}$ ,  $\Sigma_{uc}$ ,  $K_i$  et  $P_i^{-1}(\mathcal{L}(G_i))$ . Par définition, il suffit de montrer que ce langage est (i) contrôlable par rapport à  $\Sigma_{i,uc}$  et  $P_i^{-1}(\mathcal{L}(G_i))$  et (ii) contrôlable par rapport à  $\Sigma_{uc}$  et  $K_i$ . Le point (ii) est évident par définition de  $((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c}$ . Il reste donc à montrer le point (i). Soient  $s \in ((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c}$  et  $\sigma \in \Sigma_{i,uc}$  tels que  $s\sigma \in P_i^{-1}(\mathcal{L}(G_i))$ . Donc  $s\sigma \in ((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c} \cdot \Sigma_{i,uc} \cap P_i^{-1}(\mathcal{L}(G_i))$ . Or d'après (3),  $((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c} = (K \cap \mathcal{L}(G))^{\uparrow c} \cdot \Sigma_{uc}^* \cap K_i$ . Par conséquent, on considère  $s' \in (K \cap \mathcal{L}(G))^{\uparrow c}$  et  $t \in \Sigma_{uc}^*$  tels que  $s't = s$ . Donc  $s' \in L$  et  $t\sigma \in \Sigma \setminus \Sigma_s$

---

5. On rappelle que  $((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c}$  représente le plus petit langage préfixe-clos et contrôlable par rapport à  $\Sigma_{i,uc}$  et  $K_i$  contenant  $(K \cap \mathcal{L}(G))^{\uparrow c}$  (C.f. Équation [3])

(car  $\Sigma_{uc} \cap \Sigma_s = \emptyset$  et  $s't\sigma \in P_i^{-1}(\mathcal{L}(G_i))$ ). Donc d'après le lemme 1, on en déduit que  $s'P_i(t\sigma) \in \mathcal{L}(G)$ . Or  $(K \cap \mathcal{L}(G))^{\uparrow c}$  est contrôlable par rapport à  $\Sigma_{uc}$  et  $\mathcal{L}(G)$ , donc  $s'P_i(t\sigma) \in (K \cap \mathcal{L}(G))^{\uparrow c}$ . Puisque  $(K \cap \mathcal{L}(G))^{\uparrow c} \subseteq K_i$ , on en déduit que  $P_i(t\sigma) \in K_i(s')$ . Or  $t \in \Sigma_{uc}^*$  et  $\sigma \in \Sigma_{i,uc}$  donc  $P_i(t)\sigma \in K_i(s', \Sigma_{i,uc})$ . Puisque  $K$  est localement cohérent, on obtient que  $t\sigma \in K_i(s', \Sigma_{uc})$  et donc  $s\sigma (= s't\sigma) \in K_i$ . Ce qui signifie que  $s\sigma \in ((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c} \cdot \Sigma_{uc} \cap K_i$ . Or  $((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c}$  est contrôlable par rapport à  $\Sigma_{uc}$  et  $K_i$ , donc  $s\sigma \in ((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c}$ . On en déduit donc que  $((K \cap \mathcal{L}(G))^{\uparrow c})^{\downarrow K_i, c}$  est contrôlable par rapport à  $\Sigma_{uc}$  et  $P_i^{-1}(\mathcal{L}(G_i))$  et on obtient ainsi le résultat.  $\diamond$

Le théorème 4 donne donc une nouvelle condition sur le langage représentant la spécification permettant de résoudre notre problème de contrôle en n'effectuant que des calculs locaux. L'intérêt de cette nouvelle condition est qu'elle s'avère être relativement générale, permettant d'englober une classe d'objectif de contrôle importante (comme l'illustre la section suivante).

### 3.3.4. Comparaison de la cohérence locale avec la séparabilité

Dans (Willner *et al.*, 1991), les auteurs ont introduit la notion de séparabilité de langage. Lorsque cette condition est vérifiée par  $K$ , alors il existe des superviseurs locaux assurant exactement  $(K \cap \mathcal{L}(G))^{\uparrow c}$ . Ce résultat étant similaire à celui du théorème 4, il apparaît intéressant de comparer les notions de séparabilité de langages et de cohérence locale. La proposition 1 montre que la classe des langages vérifiant la condition [8] contient celle des langages séparables par rapport à  $\{\Sigma_i\}_{1 \leq i \leq n}$ .

**Proposition 1** Soit  $G = G_1 \parallel \dots \parallel G_n$  un système concurrent, avec  $\mathcal{L}(G_i) \subseteq \Sigma_i^*$ ,  $\Sigma_i = \Sigma_{i,c} \cup \Sigma_{i,uc}$  et  $\Sigma_s \subseteq \Sigma_c$ . On note  $\Sigma = \bigcup_i \Sigma_i$ ,  $\Sigma_{uc} = \bigcup_i \Sigma_{i,uc}$  et  $\Sigma_c = \bigcup_i \Sigma_{i,c}$ . Si  $K$  est séparable relativement à  $\{\Sigma_i\}_{1 \leq i \leq n}$ , alors  $K$  est localement cohérent relativement à  $\Sigma_{uc}$  et  $G$ .

**Preuve :** Soit  $K \subseteq \Sigma^*$ . Supposons que  $K$  soit séparable par rapport à  $\{\Sigma_i\}_{1 \leq i \leq n}$ . On considère,  $n$  langage  $L'_i \subseteq \Sigma_i^*$  tels que  $K = \parallel_{1 \leq i \leq n} L'_i$ . Soit  $i = \{1, \dots, n\}$ . On note  $K_i = K \cap P_i^{-1}(\mathcal{L}(G_i))$ . Soient  $s \in K_i$ ,  $s' \in K_i(s, \Sigma_{uc})$  et  $\sigma \in \Sigma_{i,uc}$  tels que  $P_i(s')\sigma \in K_i(s, \Sigma_{uc})$ . Il suffit de montrer que  $s'\sigma \in K_i(s, \Sigma_{uc})$ .

Or  $s' \in K_i(s, \Sigma_{uc})$ , donc  $ss' \in K_i$ . Donc  $\forall j \neq i$ ,  $P_j(ss') \in L'_j$ . De plus,  $\sigma \in \Sigma_i \setminus \Sigma_s$ , donc  $\forall j \neq i$ ,  $P_j(ss'\sigma) = P_j(ss')$  et on a  $P_j(ss'\sigma) \in L'_j$ . Par conséquent,  $\forall j \neq i$ ,  $ss'\sigma \in P_j^{-1}(L'_j)$ . Il suffit finalement de montrer que  $ss'\sigma \in P_i^{-1}(L'_i)$ . Or  $sP_i(s'\sigma) \in K_i$  puisque  $P_i(s')\sigma \in K_i(s, \Sigma_{uc})$  et  $P_i(s')\sigma = P_i(s'\sigma)$ . Donc  $sP_i(s'\sigma) \in K$  ce qui implique que  $sP_i(s'\sigma) \in P_i^{-1}(L'_i)$ . Finalement, on déduit du lemme 1 que  $ss'\sigma \in P_i^{-1}(L'_i)$ . D'où le résultat.  $\diamond$

La proposition 1 nous dit que sous l'hypothèse que les événements partagés sont contrôlables, la classe des langages vérifiant la condition [8] contient celle des langages séparables par rapport à  $\{\Sigma_i\}_{1 \leq i \leq n}$  (on peut en fait montrer que l'inclusion est stricte (Gaudin, 2004)). De plus, on peut noter que la vérification de la cohérence

locale d'un langage possède une complexité inférieure à celle de la vérification de la séparabilité d'un langage.

#### 4. Conclusion

Dans ce papier, nous sommes intéressés au contrôle de systèmes à événements discrets concurrents. Nous avons proposé une méthode modulaire permettant de calculer un superviseur maximal relativement à un objectif de contrôle donné  $K$  et le système  $G = G_1 \parallel \dots \parallel G_n$ . À partir de chacun des composants  $G_i$  du système  $G$ , nous commençons par dériver un nouveau système  $G_i^{-1}$ , qui s'avère en fait être une approximation du système initial. Nous assurons alors par contrôle que chaque approximation respecte une nouvelle propriété, appelée *condition de contrôlabilité partielle* relativement à l'objectif de contrôle et aux ensembles d'événements incontrôlables locaux et globaux. Assurer la contrôlabilité partielle de l'objectif de contrôle sur chacune des approximations permet, sous certaines conditions, d'en déduire un superviseur maximal assurant l'objectif de contrôle sur  $G$ .

– Une première condition correspond au cas où l'objectif de contrôle modélise un sous ensemble des comportements du système  $G$ .

– Une seconde condition, appelée *cohérence locale* garantit aussi un tel résultat (Théorème 4). Notons que la cohérence locale ne nécessite pas que l'objectif de contrôle représente un sous ensemble des comportements du système. De plus la classe des langages vérifiant cette propriété est strictement plus large que la classe des langages séparables introduites dans (Willner *et al.*, 1991)<sup>6</sup>.

L'approche adoptée dans ce papier présente plusieurs intérêts. Tout d'abord, un intérêt concernant la complexité des calculs et des systèmes manipulés : le système global n'est jamais explicité et tous les calculs sont effectués localement (i.e à partir de chacun des sous systèmes), évitant ainsi l'explosion combinatoire inhérente aux systèmes concurrents. De plus, vérifier que l'objectif de contrôle satisfait les conditions, assurant que l'approche utilisée est valide, s'effectue uniquement à partir de calculs locaux. Enfin, les calculs effectués sont réutilisables. Par conséquent, si un superviseur a été calculé et qu'un des sous-systèmes doit être remplacé par un nouveau sous-système, seuls les calculs le concernant doivent être effectués à nouveau.

*Perspectives* : Dans ce papier, seul le problème de base de la synthèse de contrôleurs a été étudié. Le résultat, i.e. le système contrôlé peut donc être bloquant. Il semble donc intéressant d'étendre notre méthodologie au cadre non-bloquant. De plus, nous avons fait l'hypothèse que les événements partagés étaient contrôlables. Or dans (Jiang *et al.*, 2000), les auteurs ont montré que cette hypothèse n'était par nécessaire avec une approche décentralisé pour montrer l'existence de superviseurs contrôlant le système global de manière à restreindre exactement celui-ci à l'objectif de contrôle. Même si au contraire de la méthode présentée dans ce papier, ce résultat n'est pas constructif, on peut se poser la question de savoir si l'hypothèse de localisation des événements incontrôlables à un composant particulier ne peut être levée. Enfin, dans un tout autre

---

6. Notons que dans (Gaudin *et al.*, 2004a), nous avons défini une condition sur l'objectif  $K$  appelée  $G$ -observabilité se situant entre la cohérence locale et la séparabilité.



ordre d'idée, nous pensons que notre méthodologie peut être appliquée à la génération automatique de tests (Jéron, 2004) pour des systèmes concurrents. Ce travail est actuellement en cours de réalisation.

## 5. Bibliographie

- Abdelwahed S., Wonham W., « Supervisory Control of Interacting Discrete Event Systems », *41th IEEE Conference on Decision and Control*, Las Vegas, USA, p. 1175-1180, 2002.
- Akesson K., Flordal H., Fabian M., « Exploiting Modularity for Synthesis and Verification of Supervisors », *Proc. of the IFAC*, July, 2002.
- Cassandras C., Lafortune S., *Introduction to Discrete Event Systems*, Kluwer Academic Publishers, 1999.
- deQueiroz M., Cury J., « Modular supervisory control of large scale discrete-event systems », *Discrete Event Systems: Analysis and Control. Proc. WODES'00*, Kluwer Academic, p. 103-110, 2000.
- Eyzell J., Cury J., « Exploiting symmetry in the synthesis of supervisors for discrete event systems », *Proc. of the American Control Conference*, Philadelphia, Pennsylvania, USA, p. 244-248, June, 1998.
- Gaudin B., Synthèse de contrôleurs sur des systèmes à événements discrets structurés, PhD thesis, Université de Rennes 1, November, 2004.
- Gaudin B., Marchand H., « Modular Supervisory Control of a class of Concurrent Discrete Event Systems », *Workshop on Discrete Event Systems, WODES'04*, September, 2004a.
- Gaudin B., Marchand H., Supervisory Control of Concurrent Discrete Event Systems, Research Report n° 1593, IRISA, February, 2004b. available at <http://www.irisa.fr/vertecs/Publis/Ps/1593.ps>.
- Jiang S., Kumar R., « Decentralized control of discrete event systems with specializations to local control and concurrent systems », *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 30, n° 5, p. 653-660, October, 2000.
- Jéron T., Contribution à la génération automatique de tests pour les systèmes réactifs, Habilitation à diriger des recherches, Université de Rennes 1, 2004.
- Leduc R., Brandin B., Wonham W., Lawford M., « Hierarchical Interface-based Supervisory Control: Serial Case », *Proc. of 40th Conf. Decision Contr.*, p. 4116-4121, 2001.
- Lee S.-H., K.C. W., « Structural Decentralized Control of Concurrent Discrete-event Systems », *European Journal of Control*, 2002.
- Minhas R., Complexity reduction in Discrete Event Systems, PhD thesis, University of Toronto, September, 2002.
- Rohloff K., Lafortune S., « The Control and Verification of Similar Agents operating in a broadcast network Environment », *42nd IEEE Conference on Decision and Control*, Hawaii, USA, December, 2003.
- Willner Y., Heymann M., « Supervisory control of concurrent discrete-event systems », *International Journal of Control*, vol. 54, n° 5, p. 1143-1169, 1991.
- Wonham W. M., Notes on Control of Discrete-Event Systems, Technical Report n° ECE 1636F/1637S, Department of ECE, University of Toronto, July, 2003.