MODULAR MULTITASKING SUPERVISORY CONTROL OF COMPOSITE DISCRETE-EVENT SYSTEMS

Max H. de Queiroz*, José E. R. Cury**

** GEMM – CEFET/SC*
Florianópolis SC 88020-301 Brazil – maxqueiroz@cefetsc.edu.br

*** LCMI – DAS – Federal University of Santa Catarina,*
Florianópolis SC 88040-900 Brazil – cury@das.ufsc.br

Abstract: A modular approach for efficiently dealing with multiple tasks and multiple specifications in the supervisory control of composite discrete-event systems (DES) is presented. Colored marking generators are used to distinguish classes of tasks in supervisory control of DES, which allows the synthesis of optimal supervisors that are nonblocking with respect to multiple control objectives. A local modular approach, combined with supervisor reduction and conflict resolution techniques, affords better efficiency to the synthesis and readability to the solution by avoiding composition of subsystems in the computation of supervisors. An example of flexible manufacturing system illustrates the proposed methodology. *Copyright © 2005 IFAC*

Keywords: Supervisory control, discrete-event systems, tasks, control system synthesis, automata, distributed control.

1. INTRODUCTION

In supervisory control theory (SCT) initiated by Ramadge and Wonham (1987), the open-loop behavior of discrete-event systems (DES) is modeled by a generator, whose marked states represent the completion of some control objective (task). The restrictions imposed on the plant can be expressed in terms of a language representing the admissible behavior. The SCT provides computational algorithms for the synthesis of a minimally restrictive supervisor that constrains the behavior of the plant by disabling controllable events in such a way that it respects the admissible language and ensures nonblocking with respect to the set of marked states. While the admissible language can be viewed as a safety specification (ensuring that nothing "bad" happens), nonblocking can be understood as a liveness specification, which ensures that the supervisor will not prevent the completion of a task (something good can happen). Situations where the liveness of multiple tasks is desired are common. In particular, interesting DES problems comprising multiple tasks arise in manufacturing and communication systems. Thistle and Malhamé (1997) and Fabian and Kumar (1997) have approached these problems by modeling the liveness specification as a set of disjunctive specifications which can be met by supervisors under given conditions (named noninterference or mutually nonblocking). In a recent paper, Queiroz *et al.* (2004) propose the use of colored marking generators – a kind of Moore automata, whose outputs identify achieved tasks – as a natural extension of SCT for the synthesis of a minimally restrictive supervisor that respects the admissible behavior and ensures the liveness of multiple tasks. In problems represented by multiple safety specifications over multiple subsystems the size of the global model grows exponentially with the composition of subsystems and thus the synthesis of supervisors for real DES may easily become unfeasible. In this paper, the modular structure of the specifications and of the plant in composite DES is exploited according to the local modular approach proposed by Queiroz and Cury (2000). By this approach, one modular supervisor can be synthesized locally for each restriction, what reduces the complexity of synthesis and affords greater flexibility to the control system. Nonetheless, even if each local modular supervisor separately ensures the liveness of all classes of tasks, the composition of supervisors could render some classes of task unreachable. This work presents conditions under which this kind of situation, named conflict, is avoided. The results in this paper are illustrated by an example of flexible manufacturing system, which is introduced in Section 2. The next section outlines the approach for modeling DES with multiple tasks as proposed by Queiroz *et al.* (2004). The following section is a summary of the framework for multitasking supervisory control. Section 5 presents the main results of this paper, which extend the multitasking control to efficiently deal with multiple restrictions in composite DES by local modular approach. Mathematical proofs can be found in the thesis of Queiroz (2004).

2. A MOTIVATING EXAMPLE

The hypothetical Flexible Manufacturing System (FMS) presented in Fig. 1 generates two types of products from raw blocks and raw pegs: a block with a conical pin on top (Product A) and a block with a cylindrical painted pin (Product B). The FMS consists of eight devices: three conveyors C_1, C_2 and C_3, a Mill, a Lathe, a Robot, a Painting Device (PD) and an Assembly Machine (AM). The devices are connected through buffers B_i, $i = 1,...,8$, with capacity for one part. The arrows in Fig. 1 indicate the flow of unfinished parts through the FMS. Raw blocks enter C1 and reach B_1. Raw pegs enter C2 and arrive in B_2. The Robot picks raw block from B_1 and places it into B_3 or moves raw pegs from B_2 to B_4. The Mill starts processing a block from B_3 and returns a geometrically shaped part with a hole on top. The Lathe can make two types of pin with the peg from B_4: a conical pin or a cylindrical pin. Then the Robot moves a finished block from B_3 to B_5, moves a conical pin from B_4 to B_6 or moves a cylindrical pin from B_4 to B_7. Conveyor 3 transports the pin from B_7 to B_8, where it is painted, and takes it back to B_7. Finally the AM creates one Product A, by assembling a block from B_5 and a conical pin from B_6, or the AM puts a finished cylindrical pin from B_7 on the top of the block, returning one Product B.

The control logic to be synthesized must give the maximal degree of freedom to the FMS while avoiding overflow or underflow of parts in the buffers and ensuring that the system always allows the manufacture of products A and B. In addition, the supervisor should not prevent the Lathe and the Mill from operating simultaneously. As further liveness specifications, the control logic should always allow the buffers B_i, $i = 1,...,6$ to become empty and the buffers B_7 and B_8 to become simultaneously empty.

In order to efficiently solve this problem, two aspects must be taken into account. First, the condition of nonblocking (and the synthesis of supervisors) must be redefined to deal with the existence of multiple control objectives that don't need to be simultaneously accomplished. Secondly, the system can be considered as a set of eight asynchronous subsystems that must be synchronized by the action of the control system in order to obey the overall specification. The specification itself can be broken into eight restrictions over the buffers, which individually affect only the respective neighboring subsystems. This fact suggests that the control system can be conceived in a modular and local fashion in spite of the monolithic synthesis, which implies the computationally problematic composition of all subsystems and specifications. Naturally, care must be taken to guarantee that the joint action of local modular supervisors preserves nonblocking of the given tasks.

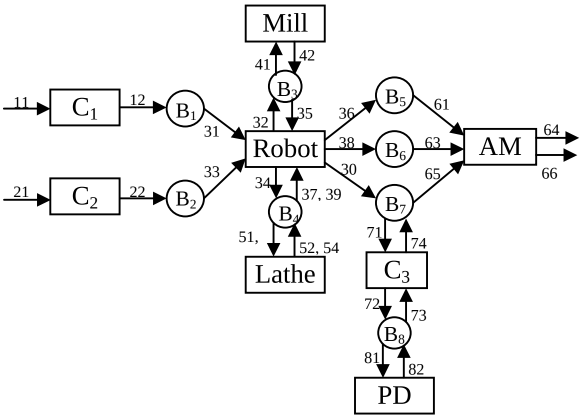

Fig. 1. Flexible manufacturing system (FMS)

3. MULTI-TASKING DES

A DES completes a task when it executes some sequence of events that accomplishes an objective of the control problem. Two tasks belong to the same class when they are related to objectives that are equivalent, i.e., that have the same meaning in the control problem. When a DES model incorporates multiple classes of tasks it is called multitasking discrete-event system (MTDES).

To distinguish multiple classes of tasks in an MTDES, a color (label) is associated to each class of task. Let Σ be the set of all events that can occur in the system and C be the set of all colors. Let Σ^* be the set of all finite strings of elements in Σ, including the empty string ε. A language L is a subset of Σ^*. $\overline{L}$ represents the prefix closure of L. Each color $c \in C$ is assigned to a language $L_c \in Pwr(\Sigma^*)$ (power set of Σ^*) that represents the set of all sequences of events in Σ that can complete a task of the respective class. Thus, the *colored behavior* of an MTDES can be modeled by the set $\Lambda_C \in Pwr(Pwr(\Sigma^*)\times C)$ given by $\Lambda_C := \{(L_c, c), c \in C\}$.

For a colored behavior Λ_C, the *language marked by* $c \in C$ is defined by $L_c(\Lambda_C) := L$ such that $(L, c) \in \Lambda_C$. The *language marked by* $B \subseteq C$ is defined by $L_B(\Lambda_C) := \cup_{b\in B} L_b(\Lambda_C)$. For $\mathrm{M}_{B1} \subseteq Pwr(Pwr(\Sigma_1^*)\times B_1)$ and $\mathrm{N}_{B2} \subseteq Pwr(Pwr(\Sigma_2^*)\times B_2)$, $\mathrm{M}_{B1} \subseteq \mathrm{N}_{B2}$ if $B_1 \subseteq B_2$ and $\forall b \in B_1$, $L_b(\mathrm{M}_{B1}) \subseteq L_b(\mathrm{N}_{B2})$. The synchronous composition of M_{B1} and N_{B2} is given by

$$\begin{aligned}\mathrm{M}_{B1} \,\|\, \mathrm{N}_{B2} := &\{(L_b(\mathrm{M}_{B1}) \,\|\, L_b(\mathrm{N}_{B2}), b), \forall b \in B_1 \cap B_2\}\\ &\cup \{(L_b(\mathrm{M}_{B1}) \,\|\, \overline{L_{B2}(\mathrm{N}_{B2})}, b), \forall b \in B_1 - B_2\}\\ &\cup \{(\overline{L_{B1}(\mathrm{M}_{B1})} \,\|\, L_c(\mathrm{N}_{B2}), b), \forall b \in B_2 - B_1\}.\end{aligned}$$

For convenience, an MTDES can be modeled by a Moore automaton, whose outputs, represented by subsets of colors, define the classes of tasks that are completed after the corresponding sequences of events. Such a state machine, named *colored marking generator* (CMG), is formally defined by a 6-tuple $G := (Q, \Sigma, C, \delta, \chi, q_0)$, where: Q is a set of states; Σ is a set of events; C is a set of colors; $\delta\colon Q \times \Sigma \to Q$ is a state transition function; $\chi\colon Q \to Pwr(C)$ is a marking function; q_0 is the initial state. For a CMG G, the *eligible event function* $\Gamma\colon Q \to Pwr(\Sigma)$ associates each state $q \in Q$ to a subset of Σ with all events that can occur in q. The *generated language* of G, denoted by $L(G)$, is the set of all finite event strings that can be reached from the initial state q_0. The marked language of an usual generator is the set of all strings that complete a task. Since a CMG usually comprises multiple classes of tasks, a marked language is defined for each class of task. Hence, the *language marked by* $c \in C$, is formally defined by $L_c(G) := \{s \in L(G) \mid c \in \chi(\delta(q_0, s))\}$. The concept of language marked by a color is extended to a nonempty set of colors as the set of event strings that complete any of the respective tasks. Then, for the color set B, $\varnothing \subset B \subseteq C$,the *language marked by B* is defined by $L_B(G) := \{s \in L(G) | \, B \cap \chi(\delta(q_0, s)) \neq \varnothing\}$. The *colored behavior* of a CMG G is given by $\Lambda_C(G) := \{(L_c(G), c), c \in C\}$.

Proposition 1: Let $\Lambda_C = \{(L_c, c), c \in C\}$ be a colored behavior. If Λ_C is regular (L_c is regular for all $c \in C$), there exists a finite state colored marking generator G such that $\Lambda_C(G) = \Lambda_C$ and $L(G) = \cup_{c\in C}\overline{L_c}$.

The *synchronous composition* of CMG G_1=$(Q_1,\Sigma_1,C_1,\delta_1,\chi_1,q_{01})$ and G_2=$(Q_2,\Sigma_2,C_2,\delta_2,\chi_2,q_{02})$, with respective eligible event functions Γ_1 and Γ_2, is defined by the CMG

$G_1 \,\|\, G_2 := Ac(Q_1{\times}Q_2,\, \Sigma_1{\cup}\Sigma_2,\, C_1{\cup}C_2,\, \delta,\, \chi,\, (q_{01},q_{02}))$, where:

$$\delta((q_1,q_2),\sigma) = \begin{cases}(\delta_1(q_1,\sigma),\delta_2(q_2,\sigma)), \text{ if } \sigma\in\Gamma_1(q_1)\cap\Gamma_2(q_2)\\ (\delta_1(q_1,\sigma),q_2), \text{ if } \sigma\in\Gamma_1(q_1)\backslash\Sigma_2\\ (q_1,\delta_2(q_2,\sigma)), \text{ if } \sigma\in\Gamma_2(q_2)\backslash\Sigma_1\\ \text{undefined otherwise;}\end{cases}$$

$$\chi((q_1,q_2)) = [\chi_1(q_1)\cup(C_2 - C_1)]\cap[\chi_2(q_2)\cup(C_1 - C_2)].$$

The operation $Ac(G)$ eliminates all unreachable states of G. The eligible event function Γ associated to $G_1 \,\|\, G_2$ is given by:

$$\Gamma((q_1,q_2)) = [\Gamma_1(q_1)\cup(\Sigma_2 - \Sigma_1)]\cap[\Gamma_2(q_2)\cup(\Sigma_1 - \Sigma_2)].$$

The open loop behavior of the FMS of Section 2 is modeled by the set of eight asynchronous CMG in Fig. 2. The manufacture of one Product A and of one Product B is respectively indicated by the tasks **a** and **b** in the model for the AM. The task **o** represents the simultaneous operation of the Lathe and the Mill. The synchronous composition of the eight CMG leads to a CMG $G = (Q, \Sigma, C, \delta, \chi, q_0)$ with 3456 states and color set $C = \{\mathbf{a}, \mathbf{b}, \mathbf{o}\}$.

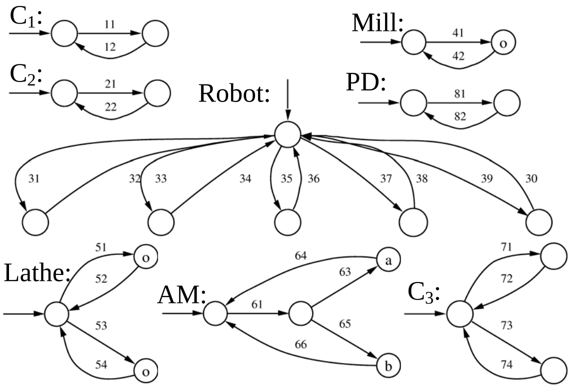

Fig. 2. Composite plant for the FMS

Given a nonempty subset of colors B, a CMG G is *strongly nonblocking w.r.t. B* if $\forall b \in B$, $L(G) = \overline{L_b(G)}$, that is, if any generated sequence of events can be completed (not necessarily in the same way) to a task of all the classes represented by colors of B. A colored behavior $\Lambda_C \in Pwr(Pwr(\Sigma^*) \times C)$ is *strongly nonblocking w.r.t.* $B \subseteq C$ when $\forall b \in B$, $\overline{L_b(\Lambda_C)} = \overline{L_C(\Lambda_C)}$.

Proposition 2: There's a CMG strongly nonblocking w.r.t. C that models a colored behavior Λ_C if and only if Λ_C is strongly nonblocking w.r.t. C.

4. MULTI-TASKING SUPERVISORY CONTROL

Let the open loop behavior (plant) of a DES be modeled by a colored marking generator $G = (Q, \Sigma, C, \delta, \chi, q_0)$, with eligible event function Γ, whose alphabet is partitioned into controllable events $\sigma \in \Sigma_c$ (events that can be disabled) and uncontrollable events $\sigma \in \Sigma_u$. Let D be a set of important tasks for which liveness (strong nonblocking) is required. Let the specification be given by a colored behavior $\mathrm{A}_D \in Pwr(Pwr(\Sigma^*) \times D)$ such that $\forall d \in D \cap C$, $\varnothing \subset L_d(\mathrm{A}_D) \subseteq L_d(G)$, and $\forall d \in E$, $\varnothing \subset L_d(\mathrm{A}_D) \subseteq L(G)$. In this way, the specification can define new classes of tasks $E = D - C$. The objective of supervisory control is to generate an entity (designated supervisor) which, by disabling controllable events, prevents the controlled system from violating the languages of the specification. The supervisor should also guarantee that the controlled system is always capable of completing tasks of D.

In the example of FMS, the controllable events are labeled with odd numbers. Each restriction can be modularly expressed by a generic specification, which is a colored behavior defined on a particular subset of events from the global alphabet of the composite plant. The generic specifications $\mathrm{M}_{gen,i}$, $i = 1,...,8$, for avoiding overflow and underflow in the buffers B_i, $i = 1,...,8$, respectively, are generated by the CMG in Fig. 3. The tasks **e*i***, $i = 1,...,6$, defined in the specifications $\mathrm{M}_{gen,i}$ indicate when the respective buffers are empty. The task **e**, indicating that buffers B_7 and B_8 are simultaneously empty, is defined by $\mathrm{M}_{gen,7}$ and $\mathrm{M}_{gen,8}$. The synchronous composition M of all generic specifications has color set $E = \{\mathbf{e1, e2, e3, e4, e5, e6, e}\}$. The global specification is then obtained from $\mathrm{A}_D = \mathrm{M} \parallel \Lambda_C(G)$. Therefore, in order to respect all the specifications defined in Section 2, the controlled system must respect A_D and be strongly nonblocking with respect to the color set $D = \{\mathbf{a, b, o, e1, e2, e3, e4, e5, e6, e}\}$.

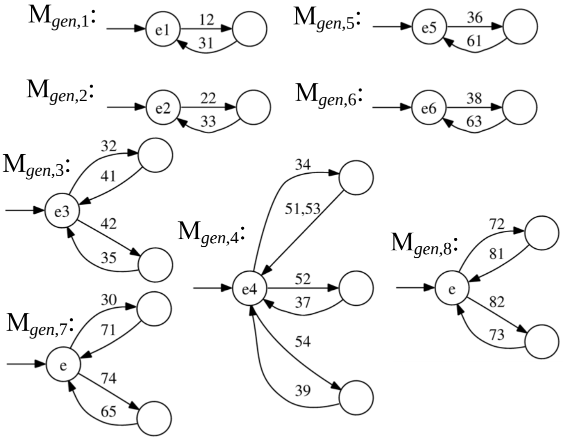

Fig. 3. CMG for the generic specifications $\mathrm{M}_{gen,i}$ respective to buffers B_i, $i = 1,...,8$

When the specifications include classes of tasks that are not explicit in the plant ($E \neq \varnothing$), the role of marking the new colors in the controlled system is assigned to the supervisor. A *coloring supervisor* is then defined as a mapping that associates to each sequence of events of the plant a set of enabled events and a set of new colors (of E) representing completed tasks. Thus, a coloring supervisor S consists of a function $S: L(G) \rightarrow Pwr(\Sigma) \times Pwr(E)$.

Let $S(s) = (\gamma, \mu)$, and write $\Re(S(s)) = \gamma$ and $\Im(S(s)) = \mu$. The events that can occur after the occurrence of a string of events $s \in L(G)$ are given by $\Re(S(s)) \cap \Gamma(\delta(q_0, s))$. A coloring supervisor S is *admissible* if it doesn't require disabling uncontrollable events, that is, if $\forall s \in L(G)$, $\Sigma_u \cap \Gamma(\delta(q_0, s)) \subseteq \Re(S(s))$. The language generated by the controlled system S/G is defined by:

1. $\varepsilon \in L(S/G)$;
2. $s\sigma \in L(S/G) \Leftrightarrow (s \in L(S/G)) \wedge (s\sigma \in L(G)) \wedge (\sigma \in \Re(S(s)))$.

The languages marked by S/G are given by:

$$\forall c \in C, L_c(S/G) := L(S/G) \cap L_c(G);$$
$$\forall e \in E, L_e(S/G) := \{s \in L(S/G) \mid e \in \Im(S(s))\}.$$

A supervisor S is *strongly nonblocking w.r.t D* if

$$\forall d \in D, \overline{L_d(S/G)} = L(S/G).$$

Theorem 1: (Queiroz *et al.*, 2004) Necessary and sufficient conditions for the existence of an admissible coloring supervisor S strongly nonblocking w.r.t. D such that $\Lambda_D(S/G) = \mathrm{A}_D$ and $L(S/G) = \overline{L_D(\mathrm{A}_D)}$ are:

1. controllability of A_D w.r.t. G: $\overline{L_D(\mathrm{A}_D)}\Sigma_u \cap L(G) \subseteq \overline{L_D(\mathrm{A}_D)}$;
2. D–closedness of A_D w.r.t. G: $\forall d \in (D \cap C), L_d(\mathrm{A}_D) = \overline{L_d(\mathrm{A}_D)} \cap L_d(G)$;
3. strong nonblocking of A_D w.r.t. D.

Next theorem (Queiroz *et al.*, 2004) proves the existence of a supremal controllable and strongly nonblocking colored behavior contained in A_D.

Theorem 2: Let the set of behaviors strongly nonblocking w.r.t. D contained in Λ_D be defined by $SNB(\mathrm{A}_D, D) := \{\mathrm{M}_D \subseteq \mathrm{A}_D \mid \forall d \in D, \overline{L_d(\mathrm{M}_D)} = \overline{L_D(\mathrm{M}_D)}\}$. $C(\mathrm{A}_D, G) := \{\mathrm{M}_D \subseteq \mathrm{A}_D \mid \overline{L_D(\mathrm{M}_D)}\Sigma_u \cap L(G) \subseteq \overline{L_D(\mathrm{M}_D)}\}$ is the set of controllable behaviors contained in A_D. The set $CSNB(\mathrm{A}_D, G, D) := C(\mathrm{A}_D, G) \cap SNB(\mathrm{A}_D, D)$ has a supremal element $SupCSNB(\mathrm{A}_D, G, D)$.

It can be shown that, for a regular A_D and a finite-state G, $SupCSNB(\mathrm{A}_D, G, D)$ can be computed from a CMG modeling A_D by the iterative elimination of states failing to be controllable or strongly nonblocking w.r.t. D. By Theorem 1, this behavior, if nonempty, can be generated by a minimally restrictive strongly nonblocking supervisor. In the example of FMS, the optimal colored behavior $SupCSNB(\mathrm{A}_D, G, D)$, computed with the help of CTCT (Wonham, 2003), can be guaranteed by a monolithic supervisor with 45504 states.

5. LOCAL MODULAR CONTROL OF MTDES

As pointed out by Queiroz and Cury (2000), it can be reasonably assumed that the plant of large-scale DES is modeled by the composition of multiple asynchronous subsystems (product system) and the global specification is given by a set of local specifications, which synchronizes subsets of subsystems. For composite DES, a monolithic control architecture implies the synthesis of supervisors from the global model of the plant, which grows exponentially with the numbers of subsystems. Therefore, it can be advantageous to design each modular supervisor on a local version of the plant, which is obtained from the composition of the subsystems affected by the respective specification. The architecture for the conjunction of local modular supervisors for multitasking composite discrete event systems, which follows the modular approach of Wonham and Ramadge (1988) and its extention for composite systems (Queiroz and Cury, 2000), is defined in the following.

Let the open-loop behavior of an MTDES be modeled by the composition of subsystems $\{G_i = (Q_i, \Sigma_i, C_i, \delta_i, \chi_i, q_{0i}),\quad i = 1,2,...,n\}$, with $\Sigma_i \cap \Sigma_j = \varnothing$, for $i \neq j$. The global alphabet is given by $\Sigma = \cup_{i = 1,...,n}\Sigma_i$. Let the specification be expressed by a set of generic admissible behaviors $\{\mathrm{M}_{gen,j} \subseteq Pwr(Pwr(\Sigma_{gen,j}^{*}) \times C_{gen,j}), j = 1,2,...,m\}$, with $\Sigma_{gen,j} \subseteq \Sigma$. For $j = 1,...,m$, the local plant $G_{loc,j} = (Q_{loc,j}, \Sigma_{loc,j}, C_{loc,j}, \delta_{loc,j}, \chi_{loc,j}, q_{0loc,j})$, with eligible event function $\Gamma_{loc,i}$, is computed by the (asynchronous) composition of all subsystems that are affected by $\mathrm{M}_{gen,j}$, i.e., that share events with $\mathrm{M}_{gen,j}$. The global plant $G = (Q, \Sigma, C, \delta, \chi, q_0)$, with eligible event function Γ, is given by $G = ||_{j = 1,2,...,m}G_{loc,j}$. For $j = 1,...,m$, the local specification $\mathrm{M}_{loc,j} \subseteq Pwr(Pwr(\Sigma_{loc,j}^{*}) \times D_{loc,j})$ is defined by $\mathrm{M}_{loc,j} := \mathrm{M}_{gen,j} \,||\, \Lambda_{Cloc,j}(G_{loc,j})$. Let the color set $E_{loc,j} := D_{loc,j} - C_{loc,j}$.

Without loss of generality, assume that $m = 2$. Let $S_{loc,1}\colon L(G_{loc,1}) \to Pwr(\Sigma_{loc,1}) \times Pwr(E_{loc,1})$ and $S_{loc,2}\colon L(G_{loc,2}) \to Pwr(\Sigma_{loc,2}) \times Pwr(E_{loc,2})$ be admissible coloring supervisors for $G_{loc,1}$ and $G_{loc,2}$, respectively. For $i = 1,2$, define the natural projections $\mathrm{P}_{loc,i}\colon \Sigma^{*} \to \Sigma_{loc,i}^{*}$. For a string $s \in L(G)$ – what implies that $\mathrm{P}_{loc,i}(s) \in L(G_{loc,i})$ –, with $S_{loc,1}(\mathrm{P}_{loc,1}(s)) = (\gamma_1, \mu_1)$ and $S_{loc,2}(\mathrm{P}_{loc,2}(s)) = (\gamma_2, \mu_2)$, the control action of the supervisor representing the conjunction of $S_{loc,1}$ and $S_{loc,2}$, $S_{loc,1} \wedge S_{loc,2}\colon L(G) \to Pwr(\Sigma) \times Pwr(E)$, where $E = E_{loc,1} \cup E_{loc,2}$, is given by

$$\begin{gathered} S_{loc,1} \wedge S_{loc,2}(s) := \\ ((\gamma_1 \cup (\Sigma_{loc,2} - \Sigma_{loc,1}))\cap(\gamma_2 \cup (\Sigma_{loc,1} - \Sigma_{loc,2})), \\ (\mu_1 \cup (E_{loc,2} - E_{loc,1}))\cap(\mu_2 \cup (E_{loc,1} - E_{loc,2}))). \end{gathered}$$

The conjunction of two local modular coloring supervisors is described in Fig. 4. The control action of local supervisors is extended to the global plant by enabling all events of Σ that are not in their own alphabet and marking all new colors of E that are not in their own color set. Thus, a shared controllable event will be enabled in the plant if and only if it is enabled by the global action of both supervisors for the strings corresponding to the sequence of events occurred in the respective local plants. In other words, it suffices for one supervisor to disable an event for it to be disabled in the plant. In the same way, the new colors that are common to the color set of both supervisors will be marked in the closed loop behavior if and only if they are marked by both supervisors. It can be shown that the closed-loop behavior under modular control is given by:

In order to deduce the properties of the controlled system in the multitasking local modular architecture from the properties of the local supervisors, the conditions under which the properties of controllability and strong nonblocking are preserved by synchronous composition are investigated. The key condition is the property of strong nonconflict, which is defined for colored behaviors with different local alphabets and color sets in the following. $\mathrm{M}_{loc,1}$ and $\mathrm{M}_{loc,2}$ are strongly nonconflicting w.r.t. $B \subseteq D$ whenever

$$\forall b \in B,\ L_b(\overline{\mathrm{M}_{loc,1}} \,||\, \overline{\mathrm{M}_{loc,2}}) = L_b(\overline{\mathrm{M}_{loc,1} \,||\, \mathrm{M}_{loc,2}}).$$

$$S_{loc,1} \wedge S_{loc,2}/G = (S_{loc,1}/G_{loc,1}) \,||\, (S_{loc,2}/G_{loc,2}).$$

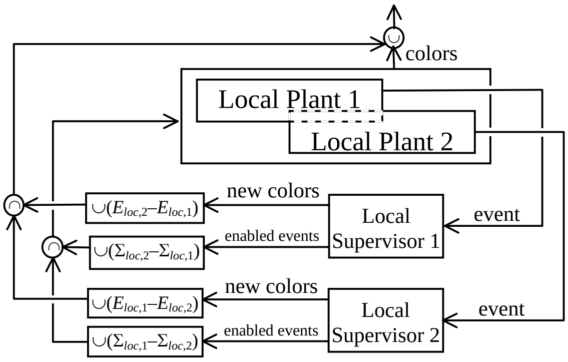
colors
Local Plant 1
Local Plant 2
new colors
$\cup(E_{loc,2}-E_{loc,1})$
enabled events
$\cup(\Sigma_{loc,2}-\Sigma_{loc,1})$
Local Supervisor 1
event
new colors
$\cup(E_{loc,1}-E_{loc,2})$
enabled events
$\cup(\Sigma_{loc,1}-\Sigma_{loc,2})$
Local Supervisor 2
event

Fig. 4. Local modular control architecture

Strong nonconflict of $\mathrm{M}_{loc,1}$ and $\mathrm{M}_{loc,2}$ w.r.t. B means that any synchronous prefix can complete any global task represented by a color of B. The following proposition indicates the relationship between strong nonconflict of colored behaviors and strong nonblocking of colored marking generators with distinct alphabets and color sets.

Proposition 3: Let $H_{loc,1}$ and $H_{loc,2}$ be strongly nonblocking (w.r.t. $D_{loc,1}$ and $D_{loc,2}$) colored marking generators for $\mathrm{M}_{loc,1}$ and $\mathrm{M}_{loc,2}$, respectively. $H_{loc,1} \,||\, H_{loc,2}$ is strongly nonblocking w.r.t. B if and only if $\mathrm{M}_{loc,1}$ and $\mathrm{M}_{loc,2}$ are strongly nonconflicting w.r.t. D.

The main results of this paper are presented below. They show that strong nonblocking and optimality of local modular control architecture can be ensured for MTDES under the condition of strong nonconflict.

Theorem 3: For a composite system defined as above, let $S_{loc,1}$ and $S_{loc,2}$ be strongly nonblocking (w.r.t. $D_{loc,1}$ and $D_{loc,2}$) admissible supervisors for $G_{loc,1}$ and $G_{loc,2}$, respectively. If $\Lambda_{D1}(S_{loc,1}/G_{loc,1})$ and $\Lambda_{D2}(S_{loc,2}/G_{loc,2})$ are strongly nonconflicting w.r.t. D, $S_{loc,1} \wedge S_{loc,2}$ is an admissible supervisor for G strongly nonblocking w.r.t. D.

Proposition 4: If $SupCSNB(\mathrm{M}_{loc,1}, G_{loc,1}, D_{loc,1})$ and $SupCSNB(\mathrm{M}_{loc,2},G_{loc,2},D_{loc,2})$ are strongly nonconflicting w.r.t. D, then $SupCSNB(\mathrm{M}_{loc,1}||\mathrm{M}_{loc,2},G,D) = SupCSNB(\mathrm{M}_{loc,1},G_{loc,1},D_{loc,1}) \,||\, SupCSNB(\mathrm{M}_{loc,2},G_{loc,2},D_{loc,2})$.

The results above can be generalized for multiple specifications by using the following extension of the definition of strong nonconflict. A set of colored behaviors $\{\mathrm{M}_i \subseteq Pwr(Pwr(\Sigma_i^{*}) \times D_i),\ i = 1,...,m\}$ is strongly nonconflicting w.r.t. $B \subseteq D$ whenever

$$\forall b \in B,\ L_b(||_{i=1,...,m}\overline{\mathrm{M}_i}) = L_b(\overline{||_{i=1,...,m}\mathrm{M}_i}\ ).$$

The results in this section are applied to the FMS as follow. For every generic specification $\mathrm{M}_{gen,i}$,

$i = 1,...,8$, its local plant $G_{loc,i}$ is obtained from the composition of all affected subsystems. The colored behaviors for the local specifications are computed by $\mathrm{M}_{loc,i} = \mathrm{M}_{gen,i} \,||\, \Lambda_{Cloc,i}(G_{loc,i})$. For $i = 1,...,8$, local modular supervisors $H_{loc,i}$ (with less than 128 states) such that $\Lambda_{Dloc,i}(H_{loc,i}) = SupCSNB(\mathrm{M}_{loc,i}, G_{loc,i}, D_{loc,i})$ are computed. Using the algorithm of Su and Wonham (2004), the sizes of all local modular supervisors were reduced to less then 5 states. The set of reduced local modular supervisors $R_{loc,i}$, $i = 1,...,8$, is illustrated in Fig. 5, where the dashed arrows indicate the events to be disabled.

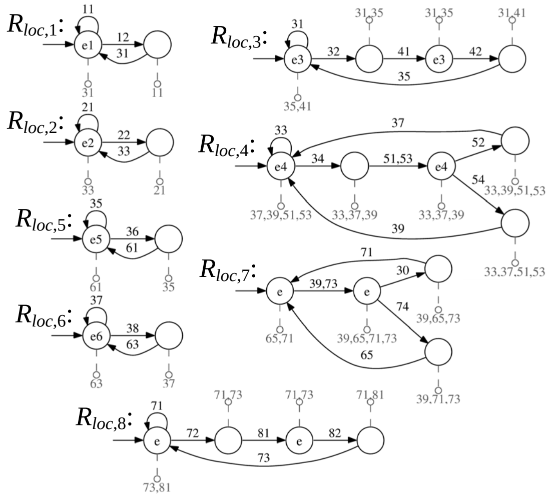

Fig. 5. Reduced local supervisors $R_{loc,i}$, $i = 1,...,8$

In order to check if the local modular supervisors are strongly nonconflicting w.r.t. D, the CMG $H = ||_{i = 1,...,8}H_{loc,i}$ (with 70272 states) is computed. By verifying that H fails to be strongly trim, it can be concluded that the modular supervision is strongly conflicting w.r.t. D. However, the synthesis can proceed by computing a coordinator that solve the global conflict. For that purpose, the blocking CMG H, representing the naive conjunction of local modular supervisors over the original plant, is taken as the new plant. The generator HS (with 45504 states) for the maximal behavior contained in H that is controllable w.r.t. H and strongly nonblocking w.r.t. D could be taken as a strongly nonblocking coordinator for H. In order to avoid the complexity of the reduction of coordinator HS, the conflict is solved in a smaller section of the FMS. By reasoning about the problem, it can be observed that the conflict that blocks the generation of Product B is related to the flow of pins. Therefore a blocking generator $H2$, with 1912 states, is computed by the composition of all supervisors related to the flow of pins: $H2 = H_{loc,4} \,||\, H_{loc,6} \,||\, H_{loc,7} \,||\, H_{loc,8}$. Then a strongly nonblocking coordinator $H2S$, with 1256 states, is obtained from $H2$. By supervisor reduction, the strongly nonblocking coordinator CS with only 2 states in Fig. 6 was obtained from $H2S$. It can be checked that $CS \,||\, H$ is equivalent to HS and thus the joint action of the set of reduced local modular supervisors with the coordinator CS is minimally restrictive and strongly nonblocking w.r.t. D.

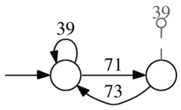

Fig. 6. Strongly nonblocking coordinator CS

6. CONCLUSION

A local modular approach has been extended to the multitasking framework to deal with state space explosion in the synthesis of supervisors for composite MTDES by exploiting modularity of plant and of specifications. The condition for optimality of the proposed architecture is strong nonconflict of locally defined colored behaviors. The simplicity of the solution for the FMS indicates that the use of multitasking local modular control combined with supervisor reduction and conflict resolution is a natural and feasible way of dealing with multiple tasks and multiple specifications in composite DES.

AKNOWLEDGEMENTS

The authors thank the financial support provided by CNPq and CAPES.

REFERENCES

Fabian, M. and R. Kumar (1997). Mutually Nonblocking Supervisory Control of DES, *CDC'97*, San Diego, CA, USA.

Queiroz, M.H. de and J.E.R. Cury (2000). Modular Supervisory Control of Large Scale DES. *In: DES: Analysis and Control*, pp. 103-110, Kluwer. (Proc. WODES 2000, Ghent, Belgium)

Queiroz, M.H. de, J.E.R. Cury and W.M. Wonham (2004). Multitasking supervisory control of DES. WODES04, Reims, France.

Queiroz, M.H. de (2004). *Controle Supervisório Modular e Multitarefa de Sistemas Compostos.* Doctor Thesis, UFSC , Florianópolis, Brazil.

Ramadge, P.J. and W.M. Wonham (1987). Supervisory control of a class of discrete event process. SIAM Journal of Control and Optimization, **25**, 206-230.

Su, R. and W.M. Wonham (2004). Supervisor Reduction for Discrete-Event Systems. *Discrete Event Dynamic Systems*, **14**, 31-53.

Thistle, J.G.R. and P. Malhamé (1997). Multiple Marked Languages and Noninterference in Modular Supervisory Control. *Proc. of the 35th Annual Allerton Conference*. pp. 523-532.

Wonham, W.M. (2003). *Notes on Control of DES*. Dept. of Elec. & Comp. Eng., Univ. of Toronto. http://www.control.utoronto.ca/DES

Wonham, W.M. and P.J. Ramadge (1988). Modular supervisory control of discrete event systems. *Math. of Control, Signals & Systems*, **1**, 13-30.