SUPERVISORY UNIQUENESS FOR OPERATING MODE SYSTEMS

Kamach Oulaid Chafik Samir Piétrac Laurent Niel Eric

Laboratoire d'Automatique Industrielle de Lyon,
Bat. Antoine de St-Exupery
27, Av. Jean Cappelle, 69621 Villeurbanne, France
Tel: (33)0472436214, Fax : (33)0472438535

Abstract: Multi-model approaches to Discrete-Event-Systems (DES) are ideally suited to implementing operating mode management and inter-mode phase alternation (switching) policy. The resulting major problem involves respecting full system evolution tracking (both plant and specifications) when inter-mode switching is evoked. In other words, after jumping from a mode to another, the newly activated mode must be directed to a state (its starting state) corresponding to the full system evolution state. The aim is therefore to determine the possible starting states of each operating mode. This study develops the underlying notion that, whilst the tracking mechanism is required at plant level, it is extended to supervision level in the sense that specification interpretation remains unchanged in relation to the various starting states. This paper attempts to demonstrate formally, using Supervisory Control Theory (SCT), that there is an unique upervisor for each operating mode by proving that all event sets authorized by the supervisor remain independent of the different starting states.

Keywords: operating modes, reactive systems, supervisory control, Discrete-Event-Systems, multi-model, switched systems.

1. INTRODUCTION

The multi-model concept involves representing a complex system by a set of simple models, each of which describes the system in a given operating mode (Kamach *et al.*, 2002). To maintain the recovery procedure, each plant level model is controlled by its proper supervisor. Problems such as alternation (or switching) and model tracking must therefore be studied. The system is, in fact, assumed to operate in a single mode, represented by its model G_i and controlled by its associated supervisor S_i. When a failure or repair event (a so-called commutation event in our context) occurs, the system will switch to another operating mode represented by its model G_j and controlled by its supervisor S_j. In this case, G_j must be directed to a state compatible with system evolution. Furthermore, the specification model of G_j must be simultaneously directed to a state compatible with the G_j model to ensure system tracking. This observation means that different starting states [1] must be considered.

This study essentially involves commutation between operating modes and demonstrates specifically conditions governing the existence of one unique supervisor for each considered operating mode, even under different starting states. Intuitively, for a given operating mode, the behavior of the resulting supervisor remains unchanged, irrespective of its starting state. This work proves formally the unity of such a supervisor.

---

[1] starting state can include initial state and other state start possibilities

Section 2 of this paper introduces selected DES multi-model design terminology and notation. Formalism applicable to the problem of commuting between designed process models is also briefly recalled in this section. Section 3 deals with the existence of supervisor conditions for each operating mode and corresponding control strategies. Study conclusions are presented in Section 4.

2. DES MULTI-MODEL DESIGN

This section focuses on guaranteed operation under failure which, whilst causing degraded production, does allow continuity of service. Reactive systems [2] are subject to failures. This type of system must be flexible to perform under controlled risks. At system design stage, this flexibility involves to taking into account different operating modes. (Kamach *et al.*, 2002) and (Kamach *et al.*, 2003) has proposed a multi-model concept, which involves designing each operating mode using just one process model. A detailed discussion dealing with the advantages of multi-model design appears in (Kamach *et al.*, 2002) and (Kamach *et al.*, 2003). We recall here only the element required for ensuring development.

To introduce the proposed approach, we consider an example involving a simple manufacturing plant. This system comprises three machines, as shown in figure 1.

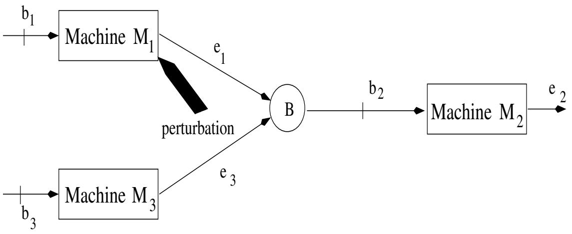

Fig. 1. Diagram of production unit example

Initially, buffer B is empty and machine M_3 is performing another task outside the unit, but it intervenes when M_1 breaks down. With event b_1 (respectively b_3), M_1 (respectively M_3) picks up a workpiece from an infinite bin and places it in buffer B, after completing its work (events e_1 respectively e_3). M_2 operates similarly, but takes its workpiece from B (event b_2) and places it in an infinite output bin, when it has finished its task (event e_2). It is assumed that only M_1 can break down (event f_1) and be repaired (event r_1) (figure 2). Two operating modes are designed for the overall system : a nominal mode G_n, in which M_1 and M_2 produce, and a degraded mode (G_d), in which M_3 replaces M_1 (figure 2). These two modes are created from models of M_1, M_2 and M_3 but they exclude f_1 and r_1 events, which are considered as inter-mode commutation events. Initially, the system is in the nominal mode described by G_n. When f_1 occurs, the system passes to the degraded mode described by G_d. Occurrence of r_1 permits transfer from G_d to G_n. This means that only one operating mode is active at any one time.

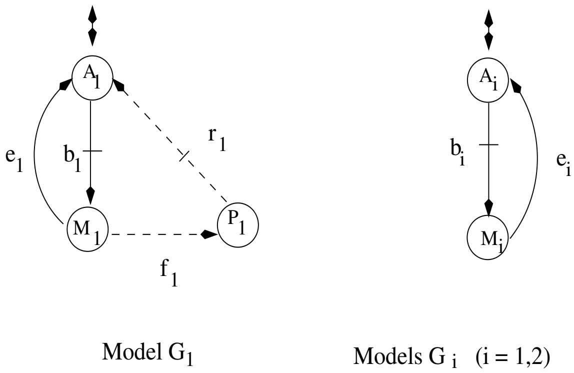

Fig. 2. Automata models of machines M_i (for $i \in \{1,2,3\}$)

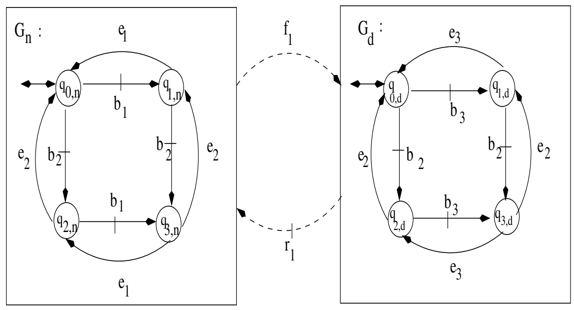

Fig. 3. Nominal and degraded process model

The objective is now to determine each operating mode along with the respective commutation conditions. To do this, let Λ as a set containing indices of all models composing the overall system with $card(\Lambda) = m < \infty$. $Card(\Lambda)$ representes the number of models to be designed. Let $\lambda_i \in \Lambda$. In the example $\Lambda = \{n, d\}$ where n is the index of the nominal mode and d the index of degraded mode. G_{λ_i} is defined as an uncontrolled automaton. Formally: $G_{\lambda_i} = (Q_{\lambda_i}, \Sigma_{\lambda_i}, \delta_{\lambda_i}, q_{0,\lambda_i}, Q_{m,\lambda_i})$, where Q_{λ_i} is a set of states, Σ_{λ_i} [3] is the set of event labels, and $\delta_{\lambda_i} : Q_{\lambda_i} \times \Sigma_{\lambda_i} \Rightarrow Q_{\lambda_i}$, the partial transition function, which is defined at each $q \in Q_{\lambda_i}$ for a subset of events $\sigma \in \Sigma_{\lambda_i}$, the initial state is q_{0,λ_i}. The marked states are $Q_{m,\lambda_i} \subseteq Q_{\lambda_i}$ and represent the end of tasks or sequences of tasks. Let $\Sigma^*_{\lambda_i}$ denote the set of all finite string over Σ_{λ_i} plus the empty strings ε. δ_{λ_i} is then extended to a function $\delta_{\lambda_i} : Q_{\lambda_i} \times \Sigma^*_{\lambda_i} \Rightarrow Q_{\lambda_i}$, such that $\forall q \in Q_{\lambda_i}$, $\delta_{\lambda_i}(q, \varepsilon) = q$ and $\delta_{\lambda_i}(q, s\sigma) = \delta_{\lambda_i}(\delta_{\lambda_i}(q,s), \sigma)$, $\sigma \in \Sigma_{\lambda_i}$, $s \in \Sigma^*_{\lambda_i}$. We can write $\delta_{\lambda_i}(q, s)!$ as an abbreviation for $\delta_{\lambda_i}(q, s)$ is defined. The language generated by G_{λ_i} is then $L(G_{\lambda_i}) := \{s \in \Sigma^*_{\lambda_i} \,|\, \delta_{\lambda_i}(q_{0,\lambda_i}, s)!\}$. In general, we assume that $\Sigma_{\lambda_i} \cap \Sigma_{\lambda_j} \neq \emptyset$ (with $i \neq j$), *i.e.*,

[2] A reactive system aims to react to failures and may lead to operating mode management

[3] Σ_{λ_i} can be partitioned to $\Sigma_{\lambda_i,c}$ and $\Sigma_{\lambda_i,uc}$ where the disjoint subsets $\Sigma_{\lambda_i,c}$ and $\Sigma_{\lambda_i,uc}$ comprise respectively the controllable and uncontrollable events.

we assume that common components can be found between two modes λ_i and λ_j. Initially the system is described by G_n. Let us define $\Sigma' = \cup_{ij}\{\alpha_{\lambda_i,\lambda_j}\}$ as the set of commutation event from G_{λ_i} to G_{λ_j}. The problem is to determine the arrival state of G_{λ_j} after the occurrence of $\alpha_{\lambda_i,\lambda_j}$ in G_{λ_i}. To do this, G_{λ_i} must be extended by adding an inactive state q_{in,λ_i} to the state set of the model G_{λ_i} so that: $G_{\lambda_i,ext} = (Q_{\lambda_i,ext}, \Sigma_{\lambda_i,ext}, \delta_{\lambda_i,ext}, q_{0,\lambda_i,ext}, Q_{m,\lambda_i,ext})$, with

- $Q_{\lambda_i,ext} = Q_{\lambda_i} \cup \{q_{in,\lambda_i}\}$,
- $\Sigma_{\lambda_i,ext} = \Sigma_{\lambda_i} \cup \Sigma'$,
- $q_{0,\lambda_i,ext} = \begin{cases} q_{0,\lambda_i} & \text{if } \lambda_i = 1 \\ q_{in,\lambda_i} & \text{if } \lambda_i \neq 1 \end{cases}$
- $Q_{m,\lambda_i,ext} = Q_{m,\lambda_i}$: marked state which equal to Q_{m,λ_i} because q_{in,λ_i} will never be marked,
- $\delta_{\lambda_i,ext}$ is defined as follows:
  (1) $\forall q \in Q_{\lambda_i}$, and $\forall \sigma \in \Sigma_{\lambda_i}$, if $\delta_{\lambda_i}(q,\sigma)!$, then $\delta_{\lambda_i,ext}(q,\sigma) := \delta_{\lambda_i}(q,\sigma)$,
  (2) $\forall q \in Q_{\lambda_i}$ from which $\alpha_{\lambda_i,\lambda_j}$ can occur (with $i \neq j$) then $\delta_{\lambda_i}(q, \alpha_{\lambda_i,\lambda_j}) = q_{in,\lambda_i}$: extended transition function allows model G_{λ_i} to be inactive if the commutation event occurs.

G_{λ_j} is similarly extended to $G_{\lambda_j,ext}$. The objective now is to define $\delta_{\lambda_j,ext}(q_{in,\lambda_j}, \alpha_{\lambda_i,\lambda_j})$. To do this, projection $\pi_{\lambda_i,\lambda_j}$ is introduced as follows:

$$\pi_{\lambda_i,\lambda_j} : (\Sigma_{\lambda_i})^* \longrightarrow (\Sigma_{\lambda_j})^* \text{ such that :}$$

$$\pi_{\lambda_i,\lambda_j}(\varepsilon) = \varepsilon$$

$$\pi_{\lambda_i,\lambda_j}(s\sigma) = \begin{cases} \pi_{\lambda_i,\lambda_j}(s)\sigma & \text{if } \sigma \in \Sigma_{\lambda_i} \cap \Sigma_{\lambda_j} \\ \pi_{\lambda_i,\lambda_j}(s) & \text{if } \sigma \in \Sigma_{\lambda_i}/\Sigma_{\lambda_j} \end{cases}$$

In other words, $\pi_{\lambda_i,\lambda_j}$ is a projection whose effect on a string $s \in \Sigma^*_{\lambda_i}$ is to erase all events σ of s that do not belong to $\Sigma_{\lambda_i} \cap \Sigma_{\lambda_j}$. This allows the behavior of common components only to be tracked. From G_{λ_j}, it allows identification of the output states of the intersection elements in G_{λ_i} when $\alpha_{\lambda_i,\lambda_j}$ occurs (*i.e.* $\alpha_{\lambda_i,\lambda_j} \in post(s\sigma)$ [4]). *E.g.* $\pi_{n,d}(b_1) = \varepsilon$ and $\pi_{n,d}(b_1 b_2) = b_2$ since $b_1 \in \Sigma_n/\Sigma_d$ et $b_2 \in \Sigma_n \cap \Sigma_d$.

2.1 Determining starting states of $G_{\lambda_i,ext}(\lambda_i \neq 1)$

Let us assume that the commutation event produced is $\alpha_{\lambda_i,\lambda_j}$ *i.e.* model $G_{\lambda_j,ext}$ must be activated. The following theorem will then give us the starting state of this model.

Theorem 2.1.
Under the foregoing assumptions, $\forall s \in L(G_{\lambda_i})$, such that $\alpha_{\lambda_i,\lambda_j} \in post(s)$. The starting state of model G_{λ_j} is given by $\delta_{\lambda_j,ext}(q_{in,\lambda_j}, \alpha_{\lambda_i,\lambda_j}) = \delta_{\lambda_j}(q_{0,\lambda_j}, \pi_{\alpha_i,\alpha_j}(s))$. ■

[4] $post(s)$ represents the next event to occur after generation of string s

Theorem 2.1 allows us to determine exactly the state to which G_{λ_j} must be directed after occurrence of $\alpha_{\lambda_i,\lambda_j}$. *E.g.* we assume that f_1 is generated after occurrence of $b_1 b_2$ in G_n. So from $q_{in,d}$, G_d can be directed to $q_{2,d}$. In fact, theorem 2.1 states that $\delta_{d,ext}(q_{in,d}, \alpha_{n,d}) = \delta_{d,ext}(q_{in,d}, f_1) = \delta_d(q_{0,d}, \pi_{n,d}(b_1 b_2)) = \delta_d(q_{0,d}, b_2) = q_{2,d}$ (since $b_1 \in \Sigma_n/\Sigma_d$ and $b_2 \in \Sigma_n \cap \Sigma_d$) (figure 4).

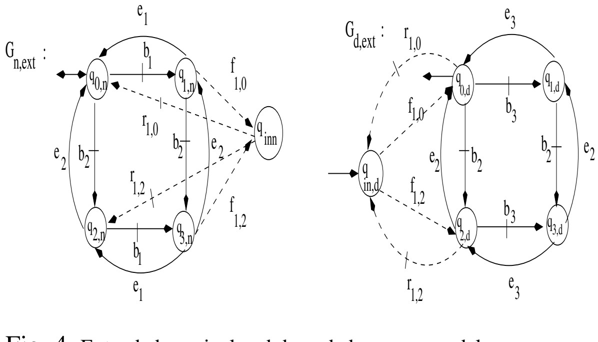

Fig. 4. Extended nominal and degraded porcess model

Since each process model has a unique inactive state, we have a nondeterministic problem. Indeed, from an inactive q_{in,λ_i}, several states can be reached for the same commutation event. To overcome this problem, we define a set of events allowing occurrences of commutation event $\alpha_{\lambda_i,\lambda_j}$: $\alpha_{\lambda_i,\lambda_j} = \alpha_{\lambda_i,\lambda_j,k}$ if $\delta_{\lambda_j}(q_{0,\lambda_j}, \pi_{\lambda_i,\lambda_j}(s)) = q_{k,\lambda_j}$ to be distinguished in model G_i (with $f_1 \in post(s)$). *E.g.* $f_1 = f_{1,2}$ if $\delta_{d,ext}(q_{0,d}, \pi_{n,d}(s)) = q_{2,d}$.

2.2 Determining of recovery states of $G_{\lambda_i,ext}$

Let us now assume $G_{\lambda_j,ext}$ is activated. Event $\alpha_{\lambda_j,\lambda_i}$ (repair event r_1 in the example) can occur. If this is the case, $G_{\lambda_j,ext}$ will be directed to its inactive state q_{in,λ_j} and $G_{\lambda_i,ext}$ will be simultaneously activated by leaving its inactive state q_{in,λ_i} to one recovery state $q \in Q_{\lambda_i}$. This state is given by applying theorem 2.2:

Theorem 2.2.
$\forall s \in L(G_{\lambda_i})$, such that $\alpha_{\lambda_i,\lambda_j} \in post(s)$ and $\forall s' \in L(G_{\lambda_j}, \delta_{\lambda_j}(q_{0,\lambda_j}, \pi_{\lambda_i,\lambda_j}(s)))$ [5], such that $\alpha_{\lambda_j,\lambda_i} \in post(s')$. Then the recovery state in model $G_{\lambda_i,ext}$ is given by:

$\delta_{\lambda_i,ext}(q_{in,\lambda_i}, \alpha_{\lambda_j,\lambda_i}) =$
$\delta_{\lambda_i}(q_{0,\lambda_i}, \pi_{\lambda_i,\lambda_j}(s)\pi_{\lambda_j,\lambda_i}(s'))$. ■

In other words, to determine the recovery state of G_{λ_j}, we must memorise the string generation history in G_{λ_i}.

In the example, commutation event r_1 can occur from states $q_{0,d}$ or $q_{2,d}$ of G_d (figure 4) assuming that f_1 has been required after occurrence of b_1 in

[5] $L(G_{\lambda_j}, \delta_{\lambda_j}(q_{0,\lambda_j}, \pi_{\lambda_i,\lambda_j}(s))) = \{s' \in \Sigma^*_{\lambda_j} \mid \delta_{\lambda_j}(\delta_{\lambda_j}(q_{0,\lambda_j}, \pi_{\lambda_i,\lambda_j}(s)), s')!\}$

G_n. From $q_{2,d}$, $\delta_{n,ext}(q_{in,n}, \alpha_{d,n}) = \delta_{n,ext}(q_{in,n}, r_1) = \delta_n(q_{0,n}, \pi_{n,d}(b_1)\pi_{d,n}(b_2e_2b_2)) = q_{2,n}$.

3. SUPERVISOR UNIQUENESS

Let G_{λ_i} and G_{λ_j} be two models of the process and suppose that G_{λ_i} is the initial model. In this case, G_{λ_i} will possess only one starting state the initial state but G_{λ_j} can possess a set $Q_{\lambda_j,st}$ of starting states q. For each $q \in Q_{\lambda_j,st}$, the behavior of G_{λ_j} is characterized by language $L(G_{\lambda_j,q}) = \{s \in \Sigma^*_{\lambda_j} \,|\, \delta_{\lambda_j}(q, s)!\}$. The interesting question is now whether there is an unique supervisor S_{λ_j} for all $G_{\lambda_j,q}$ such that: $\forall q \in Q_{\lambda_j,st}, L(S_{\lambda_j}, G_{\lambda_j,q}) = \overline{K_{\lambda_j,q}}$, where $\overline{K_{\lambda_j,q}}$ is the desired language of $G_{\lambda_j,q}$. This section discusses conditions governing the existence of such a supervisor. From (Ramadge and Wonham, 1987) there exists a supervisor S for G so that $L(S,G) = \overline{K}$ if and only if $\overline{K}$ is controllable. That is $\overline{K}\Sigma_{uc} \cap L(G) \subseteq \overline{K}$.

Let $\overline{K_{\lambda_i}}$ be the desired language of G_{λ_i}. $\{\overline{K_{\lambda_j,q}} \,|\, q \in Q_{\lambda_j,st}\}$ is the set of desired languages respectively for $\{G_{\lambda_j,q} \,|\, q \in Q_{\lambda_j,st}\}$. The objective here is to show that there is also a single supervisor S_{λ_j} for $G_{\lambda_j,q}$ whatever $q \in Q_{\lambda_j,st}$. Theorem 3.1 states necessary and sufficient conditions for the existence of a such supervisor.

Theorem 3.1.
Let G_{λ_j} be an automaton with $m > 1$ starting states $q \in Q_{\lambda_j,st}$ and $\{\overline{K_{\lambda_j,q}} \,|\, q \in Q_{\lambda_j,st}\}$ a set of possible desired languages of G_{λ_j}. Supervisory control S_{λ_j} exists such that $\forall q \in Q_{\lambda_j,st}, L(S_{\lambda_j}, G_{\lambda_j,q}) = \overline{K_{\lambda_j,q}}$ if and only if:

(1) $\forall q \in Q_{\lambda_j,st}$, $\overline{K_{\lambda_j,q}}$ is controllable w.r.t. $L(G_{\lambda_j,q})$,
(2) $\forall (q,q') \in Q_{\lambda_j,st} \times Q_{\lambda_j,st}$, $(\forall s \in \overline{K_{\lambda_j,q}}, s' \in \overline{K_{\lambda_j,q'}})$ and $s = s'$, $(\forall \sigma \in \Sigma_{\lambda_j,c})$, if $s\sigma \in \overline{K_{\lambda_j,q}}$, such that $s'\sigma \in L(G_{\lambda_j,q'})$, then $s'\sigma \in \overline{K_{\lambda_j,q'}}$,
(3) condition 2 holds with s and s' interchanged *i.e.* if $s'\sigma \in \overline{K_{\lambda_j,q'}}$, then $s\sigma \in \overline{K_{\lambda_j,q}}$. ■

Condition 1 of theorem 3.1 shows that controllability is a necessary but not a sufficient condition for supervisory control of a multi-model DES. Conditions 2 and 3 show that if an event σ is enabled by S_{λ_j} while the starting state of G_{λ_j} is q, and σ is also possible from state $q' \in Q_{\lambda_j,st}$, then σ must be enabled by S_{λ_j}. Note that the purpose of theorem 3.1 is to show that by using basic supervisory control for a multi-model DES, only one supervisor S_{λ_j}, $\forall q \in Q_{\lambda_j,st}$, can be designed such that $L(S_{\lambda_j}, G_{\lambda_j,q}) = \overline{K_{\lambda_j,q}}$. However, in conventional supervisory control, plant models possess only one initial state. To prove theorem 3.1, we extend G_{λ_j} to $G_{\lambda_j,ext}$ possessing only one initial state. In this case conventional SCT can be applied. The followinge 2 stages are required to achieve this.

(1) Extend first the model of G_{λ_j} to $G_{\lambda_j,ext} = (Q_{\lambda_j,ext}, \Sigma_{\lambda_j,ext}, \delta_{\lambda_j,ext}, q_{0,\lambda_j,ext}, Q_{m,\lambda_j,ext})$ as described in section 2 to obtain a model with only one starting state q_{in,λ_j}. This is then the unique initial state of $G_{\lambda_j,ext}$. We can then design a supervisor using a conventional supervisory control approach,
(2) Extend also $\overline{K_{\lambda_j,q}}$ by adding a commutation event $(\alpha_{\lambda_j,\lambda_i})_q$ such that $\overline{K_{\lambda_j,q,ext}} := \overline{(\alpha_{\lambda_i,\lambda_j})_q K_{\lambda_j,q}} = \{s \,|\, \exists v \in \Sigma^*_{\lambda_j}, sv \in (\alpha_{\lambda_i,\lambda_j})_q K_{\lambda_j,q}\}$. $(\alpha_{\lambda_i,\lambda_j})_q$ is the commutation event from G_{λ_i} to G_{λ_j} when the starting state of G_{λ_j} is q.

Now, from $\{\overline{K_{\lambda_j,q}} \,|\, q \in Q_{\lambda_j,st}\}$ we can determine the unique corresponding desired language for $G_{\lambda_j,ext}$. Let this language be $\overline{\cup_{q \in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$.

We now try to show that there is a supervisor $S_{\lambda_j,ext}$ such that $L(S_{\lambda_j,ext}, G_{\lambda_j,ext}) = \overline{\cup_{q \in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ if and only if $\overline{\cup_{q \in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is controllable. If $S_{\lambda_j,ext}$ exists, it will observe all the event of $\Sigma_{\lambda_j,ext}$. We try to prove the existence of S_{λ_j} that observing only the events of Σ_{λ_j}. For this, we introduce the projection function P_{λ_j} defined as follows:

$$P_{\lambda_j} : (\Sigma_{\lambda_j,ext})^* \longrightarrow (\Sigma_{\lambda_j})^* \text{ such that :}$$

$$P_{\lambda_j}(\varepsilon) = \varepsilon$$

$$P_{\lambda_j}(s\sigma) = \begin{cases} P_{\lambda_j}(s)\sigma & \text{if } \sigma \in \Sigma_{\lambda_j} \\ P_{\lambda_j}(s) & \text{otherwise} \end{cases}$$

Let $S_{\lambda_j} : P_{\lambda_j}(\Sigma^*_{\lambda_j,ext}) \Rightarrow \Gamma := \{\gamma \in Pwr(\Sigma_{\lambda_j,ext}) : \Sigma_{\lambda_j,ext,uc} \subseteq \gamma\}$, ($\Sigma_{\lambda_j,ext,uc}$ is the set of uncontrollable events), such that $\forall s \in \Sigma^*_{\lambda_j,ext}, S_{\lambda_j}(P_{\lambda_j}(s)) = S_{\lambda_j,ext}(s)$.
Knowing that
$L(S_{\lambda_j,ext}, G_{\lambda_j,ext}) = \overline{\cup_{q \in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ and

$S_{\lambda_j}(P_{\lambda_j}(s)) = S_{\lambda_j,ext}(s)$, then
$L(S_{\lambda_j}, G_{\lambda_j,ext}) = \overline{\cup_{q \in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$
if and only if:

(1) $\overline{\cup_{q \in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is controllable w.r.t $L(G_{\lambda_j,ext})$,
(2) $\overline{\cup_{q \in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is observable w.r.t $L(G_{\lambda_j,ext})$ and P_{λ_j} ((Rudie and Wonham, 1992) and (Jiang and Kumar, 2000)).

If these two conditions are validated, we can state that there is then one unique supervisor S_{λ_j} for G_{λ_j} such that $\forall q \in Q_{\lambda_j,st}, L(S_{\lambda_j}, G_{\lambda_j,q}) = \overline{K_{\lambda_j,q}}$.

To prove theorem 3.1, it is helpful to introduce the following lemmas. Thereafter we consider the following notation $K_{\lambda_j,q,ext} := (\alpha_{\lambda_i,\lambda_j})_q K_{\lambda_j,q}$.

Lemma 3.2.

$\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}} = \cup_{q\in Q_{\lambda_j,st}} \overline{K_{\lambda_j,q,ext}}$. ■

Lemma 3.3.
$\forall q \in Q_{\lambda_j,st}$, $(\alpha_{\lambda_i,\lambda_j})_q\overline{K_{\lambda_j,q}} = \overline{(\alpha_{\lambda_i,\lambda_j})_q K_{\lambda_j,q}}$. ■

Lemma 3.4.
$\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is controllable w.r.t. $L(G_{\lambda_j,ext})$ if and only if $\forall q \in Q_{\lambda_j,st}$, $\overline{K_{\lambda_j,q}}$ is controllable w.r.t. $L(G_{\lambda_j,q})$. ■

Proof of Lemma 3.2
See (Ramadge and Wonham, 1987).
Proof of Lemma 3.3

(1) First show that $\forall q \in Q_{\lambda_j,st}$, $(\alpha_{\lambda_i,\lambda_j})_q\overline{K_{\lambda_j,q}} \subseteq \overline{(\alpha_{\lambda_i,\lambda_j})_q K_{\lambda_j,q}}$.

Let $s \in (\alpha_{\lambda_i,\lambda_j})_q\overline{K_{\lambda_j,q}} \Rightarrow s = (\alpha_{\lambda_i,\lambda_j})_q u$ with $u \in \overline{K_{\lambda_j,q}}$, then $\exists v \in \Sigma^*_{\lambda_j} \,|\, uv \in K_{\lambda_j,q}$ this means that $(\alpha_{\lambda_i,\lambda_j})_q uv \in (\alpha_{\lambda_i,\lambda_j})_q K_{\lambda_j,q}$

$\Rightarrow s = (\alpha_{\lambda_i,\lambda_j})_q u \in \overline{(\alpha_{\lambda_i,\lambda_j})_q K_{\lambda_j,q}}$
$\Rightarrow s = (\alpha_{\lambda_i,\lambda_j})_q u \in \overline{K_{\lambda_j,q,ext}}$.

(2) Now schow that $\overline{(\alpha_{\lambda_i,\lambda_j})_q K_{\lambda_j,q}} \subseteq (\alpha_{\lambda_i,\lambda_j})_q\overline{K_{\lambda_j,q}}$.
Let $s \in \overline{(\alpha_{\lambda_i,\lambda_j})_q K_{\lambda_j,q}}$, then $\exists u$ and $s' \in \Sigma^*_{\lambda_j}$ (with $s = (\alpha_{\lambda_i,\lambda_j})_q s'$) such that
$(\alpha_{\lambda_i,\lambda_j})_q s'u \in (\alpha_{\lambda_i,\lambda_j})_q K_{\lambda_j,q}$.
So $s'u \in K_{\lambda_j,q}$,
then $s' \in \overline{K_{\lambda_j,q}} \Rightarrow (\alpha_{\lambda_i,\lambda_j})_q s' \in (\alpha_{\lambda_i,\lambda_j})_q\overline{K_{\lambda_j,q}}$.
Hence $s \in (\alpha_{\lambda_i,\lambda_j})_q\overline{K_{\lambda_j,q}}$

of lemma 3.4

(1) Suppose that $\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is controllable w.r.t $L(G_{\lambda_j,ext})$ and show that $\overline{K_{\lambda_j,q}}$ is controllable w.r.t. $L(G_{\lambda_j,q})$.
Let $s \in \overline{K_{\lambda_j,q}}$
$\Rightarrow (\alpha_{\lambda_i,\lambda_j})_q s \in (\alpha_{\lambda_i,\lambda_j})_q\overline{K_{\lambda_j,q}}$
$\Rightarrow (\alpha_{\lambda_i,\lambda_j})_q s \in \overline{(\alpha_{\lambda_i,\lambda_j})_q K_{\lambda_j,q}}$(lemma 3.3)
$\Rightarrow (\alpha_{\lambda_i,\lambda_j})_q s \in \cup_{q\in Q_{\lambda_j,st}} \overline{(\alpha_{\lambda_i,\lambda_j})_q K_{\lambda_j,q}}$
$\Rightarrow (\alpha_{\lambda_i,\lambda_j})_q s \in \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ (lemma 3.2).
In other words, let $\sigma \in \Sigma_{\lambda_j,uc}$ such that $s\sigma \in L(G_{\lambda_j,q})$. However, $\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is controllable w.r.t. $L(G_{\lambda_j,ext})$, it follows that $(\alpha_{\lambda_i,\lambda_j})_q s\sigma \in \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ (by controllability)
$\Rightarrow (\alpha_{\lambda_i,\lambda_j})_q s\sigma \in \cup_{q\in Q_{\lambda_j,st}} \overline{K_{\lambda_j,q,ext}}$
$\Rightarrow (\alpha_{\lambda_i,\lambda_j})_q s\sigma \in \cup_{q\in Q_{\lambda_j,st}} (\alpha_{\lambda_i,\lambda_j})_q\overline{K_{\lambda_j,q}}$
$\Rightarrow s\sigma \in \overline{K_{\lambda_j,q}}$,

(2) Now suppose that $\forall q \in Q_{\lambda_j,st}$, $\overline{K_{\lambda_j,q}}$ is controllable w.r.t. $L(G_{\lambda_j,q})$ and show that
$\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is controllable w.r.t. $L(G_{\lambda_j,ext})$.
$\forall q \in Q_{\lambda_j,st}$, $\overline{K_{\lambda_j,q}}$ is controllable w.r.t. $L(G_{\lambda_j,q})$
means that $\forall q \in Q_{\lambda_j,st}$,
$(\alpha_{\lambda_i,\lambda_j})_q\overline{K_{\lambda_j,q}} = \overline{K_{\lambda_j,q,ext}}$ (lemma 3.3) is controllable w.r.t $L(G_{\lambda_j,ext})$ because the commutation event $(\alpha_{\lambda_i,\lambda_j})_q$ is always enabled by $S_{\lambda_j,ext}$.
Thus $\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is controllable w.r.t. $L(G_{\lambda_j,ext})$ as required because controllability is preserved under unions.

Proof of theorem 3.1.
We have seen that:
$L(S_{\lambda_j,ext}, G_{\lambda_j,ext}) = \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ if and only if $\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is controllable. From ((Lin and Wonham, 1988)). We can also see that: $\forall q \in Q_{\lambda_j,st}$, $L(S_{\lambda_j}, G_{\lambda_j,ext}) = \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$, *i.e.* $\forall \sigma \in (\Sigma_{\lambda_j,ext} - \Sigma_{\lambda_j})$[6], then σ is always enabled by S_{λ_j}. On the other hand, there is a supervisor S_{λ_j} such that: $L(S_{\lambda_j}, G_{\lambda_j,ext}) = \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$, if and only if

- $\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is controllable w.r.t $L(G_{\lambda_j,ext})$,
- $\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is observable w.r.t $L(G_{\lambda_j,ext})$ and P_{λ_j}.

If these two conditions are satisfied, then there is an unique supervisor S_{λ_j} such that
$\forall q \in Q_{\lambda_j,st}, L(S_{\lambda_j}, G_{\lambda_j}) = \overline{K_{\lambda_j,q}}$.
Note that this observation is equivalent to conditions 2 and 3 of theorem 3.1.
1. Controllability
Suppose that $S_{\lambda_j,ext}$ exists, then $\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is controllable w.r.t $L(G_{\lambda_j,ext})$. Now if
$\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is controllable w.r.t $L(G_{\lambda_j,ext})$, then $\forall q \in Q_{\lambda_j,st}$, $\overline{K_{\lambda_j,q}}$ is also controllable w.r.t. $L(G_{\lambda_j,q})$ (Lemma 3.4).
Hence if $L(S_{\lambda_j}, G_{\lambda_j,ext})$ is controllable w.r.t $L(G_{\lambda_j,ext})$, then $L(S_{\lambda_j}, G_{\lambda_j,q})$ is also controllable w.r.t. $L(G_{\lambda_j,q})$ as required.
2. Observability
We must now demonstrate the equivalence relationship between conditions 2 and 3 of theorem 3.1 and observability of $\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$. Note that $\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is observable w.r.t $L(G_{\lambda_j,ext})$ and P_{λ_j} if ($\forall \sigma \in \Sigma_{\lambda_j,c}$), $\forall s, s' \in \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$, $P_{\lambda_j}(s) = P_{\lambda_j}(s')$ and $s\sigma \in \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$, $s'\sigma \in L(G_{\lambda_j,ext})$, then $s'\sigma \in \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$.
A) Suppose that $\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is observable w.r.t $L(G_{\lambda_j,ext})$ and P_{λ_j}. Now $\forall (q, q') \in Q_{\lambda_j,st} \times Q_{\lambda_j,st}$, let $\sigma \in \Sigma_{\lambda_j,c}$, $s \in \overline{K_{\lambda_j,q}}$ and $s' \in \overline{K_{\lambda_j,q'}}$ such that $s = s'$.

[6] $(\Sigma_{\lambda_j,ext} - \Sigma_{\lambda_j}) = \{\sigma \in \Sigma_{\lambda_j,ext} \,|\, \sigma \notin \Sigma_{\lambda_j}\}$

If $s\sigma \in \overline{K_{\lambda_j,q}}$ and $s'\sigma \in L(G_{\lambda_j,q'})$, one must show that $s'\sigma \in \overline{K_{\lambda_j,q'}}$ (condition 2 of theorem 3.1).
$s \in \overline{K_{\lambda_j,q}}$
$\Rightarrow (\alpha_{\lambda_i,\lambda_j})_q s \in (\alpha_{\lambda_i,\lambda_j})_q \overline{K_{\lambda_j,q}}$
$\Rightarrow (\alpha_{\lambda_i,\lambda_j})_q s \in \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$.
In other words, $s' \in \overline{K_{\lambda_j,q'}}$
$\Rightarrow (\alpha_{\lambda_i,\lambda_j})_{q'} s' \in \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$.
Since $s = s'$, $P_{\lambda_j}((\alpha_{\lambda_i,\lambda_j})_q s) = P_{\lambda_j}((\alpha_{\lambda_i,\lambda_j})_{q'} s')$.
On the other hand
$(\alpha_{\lambda_i,\lambda_j})_q s\sigma \in \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$
$\Rightarrow (\alpha_{\lambda_i,\lambda_j})_{q'} s'\sigma \in \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$
because $\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is observable.
$\Rightarrow (\alpha_{\lambda_i,\lambda_j})_{q'} s'\sigma \in \cup_{q\in Q_{\lambda_j,st}} \overline{K_{\lambda_j,q,ext}}$(lemma 3.2)
$\Rightarrow (\alpha_{\lambda_i,\lambda_j})_{q'} s'\sigma \in \overline{K_{\lambda_j,q',ext}}$
$\Rightarrow (\alpha_{\lambda_i,\lambda_j})_{q'} s'\sigma \in (\alpha_{\lambda_i,\lambda_j})_{q'} \overline{K_{\lambda_j,q'}}$
$\Rightarrow s'\sigma \in \overline{K_{\lambda_j,q'}}$.
Condition 2 of theorem 3.1 is now checked. Condition 3 holds with s and s' interchanged.
B) Supposing that condition 2 of theorem 3.1 is true and $P_{\lambda_j}(s) = P_{\lambda_j}(s')$. One must show that if $s\sigma \wedge s' \in \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ and $s'\sigma \in L(G_{\lambda_j,ext})$, then $s'\sigma \in \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ (observation).
Let $s\sigma \in \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$. This means that $\exists v \in \Sigma^*_{\lambda_j} \,|\, s\sigma v \in \cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext} \Rightarrow \exists q \in Q_{\lambda_j,st}$ such that $s\sigma v \in K_{\lambda_j,q,ext}$. So $s\sigma v = (\alpha_{\lambda_i,\lambda_j})_q s_1 \sigma v \in K_{\lambda_j,q,ext}$ (with $s = (\alpha_{\lambda_i,\lambda_j})_q s_1) \Rightarrow s_1\sigma v \in K_{\lambda_j,q} \Rightarrow s_1\sigma \in \overline{K_{\lambda_j,q}}$. On the other hand $s' \in \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$, then ther $\exists t \in \Sigma^*_{\lambda_j} \,|\, s't \in \cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}$, namely there $\exists q' \in Q_{\lambda_j,st}$ such that $s't \in K_{\lambda_j,q',ext} \Rightarrow s't = (\alpha_{\lambda_i,\lambda_j})_{q'} s_2 t \in K_{\lambda_j,q',ext}$ (with $s' = (\alpha_{\lambda_i,\lambda_j})_{q'} s_2) \Rightarrow s_2 t \in K_{\lambda_j,q'} \Rightarrow s_2 \in \overline{K_{\lambda_j,q'}}$. Furthermore $s_1 = s_2$ since $P_{\lambda_j}(s) = P_{\lambda_j}(s')$. However from condition 2 of theorem 3.1 $s_2\sigma \in \overline{K_{\lambda_j,q'}}$. Hence $(\alpha_{\lambda_i,\lambda_j})_{q'} s_2\sigma = s'\sigma \in (\alpha_{\lambda_i,\lambda_j})_{q'} \overline{K_{\lambda_j,q'}}$. So $s'\sigma \in \overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$. Consequently if conditions 2 and 3 of theorem 3.1 are true, then $\overline{\cup_{q\in Q_{\lambda_j,st}} K_{\lambda_j,q,ext}}$ is observable w.r.t
$(L(G_{\lambda_j,q,ext}), P_{\lambda_j})$. So for DES multi-modelling $\forall q \in Q_{\lambda_j,st}$, only one unique supervisor S_{λ_j} (if it exists) controls some operating mode described by its model G_{λ_j}. ■

4. CONCLUSION

Research described in this paper has been performed within the context of designing and controlling a multi-model for a Discrete Event System. When alternations are required, such systems need to be tracked from one operating mode to another. This problem has been presented and previously solved in and it is briefly recalled here. A major requirement was to study conditions governing the existence of a unique supervisory controller (if one exists) for each operating mode (irrespective of its starting state). It has been proved that the property of controllability is a necessary, but not a sufficient, condition. The property of observation has been included to complement this existence condition.

REFERENCES

Jiang, S. and R. Kumar (2000). Decentralized control of discrete event systems with specializations to local control and concurrent systems. *IEEE Transactions on Systems, Man and Cybernetics, Part B* **30**, 653–660.

Kamach, O., L. Piétrac and E. Niel (2003). Multi-model approach to discrete events systems : application to operating mode management. In: *Imacs Multiconference Computational Engineering in Systems Applications (CESA)*. Ecole Centrale de lille. reference S2-R-00-0315.

Kamach, O., S. Chafik, L. Piétrac and E. Niel (2002). Representation of a reactive systems with different models. In: *IEEE International Conference on Systems Man and Cybernetics (SMC)*. Hammamet, Tunisie. référence TA2L4 sur CDROM.

Lin, F. and W. Wonham (1988). Decentralised supervisory control of discrete event systems. *Information sciences* **25**, 1202–1218.

Ramadge, P. and W. Wonham (1987). Supervisory control of a class of discrete event processes. *SIAM Journal of Control and optimisation* **25**, 206–230.

Rudie, K. and W. Wonham (1992). hink globally act locally:decentralized supervisory control. *IEEE transactions on automatics and control* **37**, 1692–1708.