Representation of a reactive system with different models

O. Kamach, S. Chafik, L. Piétrac and E. Niel

Laboratoire d'Automatique Industrielle

Institut National des Sciences Appliquées

Bat St Exupéry – 25 av Jean Cappelle – 69621 Villeurbanne CEDEX - France

okamach@lai.insa-lyon.fr

Abstract-- **In this paper, we propose an approach which considers different models of a process (multi-model approach) based on the supervision theory of Ramadge and Wonham (RW) [1] [2]. Our contribution enables us to take into account various models which represent different operating modes of the process. In this approach only modes that ensure the same operating mode are actives while the others must be put into their respective inactive state. The problem of commutation between all designed models is formalised by a proposed framework which allows to determine each model and the commutation conditions.**

Keywords: modelling, reactive systems, operating modes, discrete event systems.

I. Introduction

In the supervisory control theory of Ramadge and Wonham [1] [2] [3] some control theory problems, such as synthesis of controlled dynamic invariant system by feedback, and concepts such as controllability and non blocking have been investigated. However, in this theory the plant is often a product of a number of simple components. Thus its state size increases exponentially with the number of components and synthesising a controller becomes laborious.

A standard way to handle state explosion is by decentralised control. This approach consists of decomposing a system to be controlled G into subsystems G_i [5] [6] [7] for which local supervisors are fairly easy to obtain. Furthermore, reactive systems are subject to failures. This type of systems must be flexible in order to behave under controlled risks and allowing continuity of service which represents the prime aim of this paper. Flexibility is expressed by different operating modes of the system. In this paper a decentralised approach is used to model each operating mode and strategy commutation from an operating mode to another one. In our case only one operating mode is active at the some time. A framework is proposed to ensure the commutation.

II. Modelling of a multi-model reactive system

Guaranteed functioning under failure causing downgraded production, yet allowing continuity of service, represents the prime aim of this section.

Reactive systems are subject to failures. This type of system must be flexible in order to behave under controlled risks. This flexibility is expressed by different operating modes of the system. In this section we are interested in the modelling of these operating modes by applying a multi-model concept which consists of designing a model process for each operating mode. The problem of commutation between all designed models is formalised by a proposed framework.

To introduce this formal framework, we consider a simple example and we will be limited to two models of the system. In figure 1.b two different models of a global system (Unit production) are represented. This system is composed of three machines as shown in fig. 1.

Initially the buffer is empty and M_3 is carrying out another task outside the unit but which intervenes when M_1 breaks down. With the event b_1, M_1 takes a workpiece from an infinite bin and enters q_1 state but deposits it in the buffer B after completing its work. M_2 operates similarly, but takes its workpiece from B and deposits it when finished in an infinite output bin.

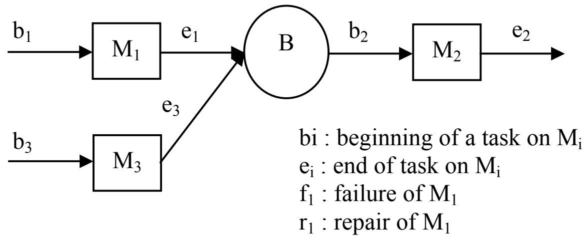


Fig. 1. Scheme of Production unit

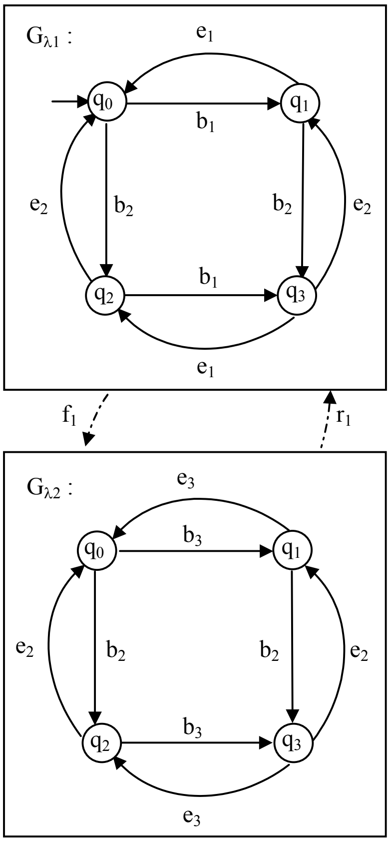


Fig. 2. Two possible models of the production unit

Now the aim is to determine each model and the commutation conditions. For this, we define Λ as a set containing indices of all models composing the global system with $\mathrm{card}(\Lambda) = n < \infty$.
$\mathrm{Card}(\Lambda)$ represents the number of models to be designed. In our case $\Lambda := \{\lambda_1, \lambda_2\}$ so $\mathrm{card}(\Lambda) = 2$.
Let $\lambda_i \in \Lambda$, we define $G_{\lambda i}$ as an uncontrollable DES which is taken to be an automaton of the model λ_i

$$G_{\lambda i} = (Q_{\lambda i},\ \delta_{\lambda i},\ \Sigma_{\lambda i},\ q_{0,\lambda i},\ Q_{m,\lambda i})^1.$$

We suppose that $\Sigma_{\lambda i} \cap \Sigma_{\lambda j} \neq \varnothing$ ($i \neq j$) and initially the process model is $G_{\lambda 1}$. Let $\Sigma' = \cup_{ij} \{\alpha_{\lambda i,\lambda j}\}$ with $\alpha_{\lambda i,\lambda j}$ represents the commutation events from $G_{\lambda i}$ to $G_{\lambda j}$. In our example $\Sigma' = \{f_1, r_1\}$. At the occurrence of $\alpha_{\lambda i,\lambda j}$ the process model becomes $G_{\lambda j}$. However, in this case, we must determine the reception state of $G_{\lambda j}$ after the commutation. To do this, we extend $G_{\lambda i}$ and $G_{\lambda j}$ by adding respectively an inactive state $q_{in,\lambda i}$ to the state set of the model $G_{\lambda i}$ and an inactive state $q_{in,\lambda j}$ to $G_{\lambda j}$ state set. At the occurrence of $\alpha_{\lambda i,\lambda j}$, $G_{\lambda i}$ will be lead to $q_{in,\lambda i}$ and $G_{\lambda j}$ will be activated from $q_{in,\lambda j}$. However, the problem is to determine the arrival state of $G_{\lambda j}$ (respectively $G_{\lambda i}$) at the occurrence of $\alpha_{\lambda i,\lambda j}$ (respectively $\alpha_{\lambda j,\lambda i}$).

[1] $Q_{\lambda i}$: Set of states, $\Sigma_{\lambda i}$: the set of alphabet, $\delta_{\lambda i}$: the function transition, $q_{0,\lambda i}$: the initial state and $Q_{m,\lambda i}$: the Set of marked states.

Let $G_{\lambda i,ext} = (Q_{\lambda i,ext},\ \delta_{\lambda i,ext},\ \Sigma_{\lambda i,ext},\ q_{0,\lambda i,ext},\ Q_{m,\lambda i,ext})$ with :

- $Q_{\lambda i,ext} = Q_{\lambda i} \cup \{q_{in,\lambda i}\}$
- $\Sigma_{\lambda i,ext} = \Sigma_{\lambda i} \cup \Sigma'$
- $q_{0,\lambda i,ext} = q_{0,\lambda i}$
- $Q_{m,\lambda i,ext} = Q_{m,\lambda i}$
- $\delta_{\lambda i,ext}$ is defined as follows :
  1. $\forall q \in Q_{\lambda i}$ and $\forall \sigma \in \Sigma_{\lambda i}$, if $(\delta_{\lambda i}(q, \sigma)!)^2$, 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, then $\delta_{\lambda i,ext}(q, \alpha_{\lambda i,\lambda j}) = q_{in,\lambda i}$.

$\delta_{\lambda i,ext}(q_{in,\lambda i}, \alpha_{\lambda j,\lambda i})$ will be defined later.

Now let us define $G_{\lambda j,ext}$ to be the extended model of $G_{\lambda j}$.
$G_{\lambda j,ext} = (Q_{\lambda j,ext},\ \delta_{\lambda j,ext},\ \Sigma_{\lambda j,ext},\ q_{0,\lambda j,ext},\ Q_{m,\lambda j,ext})$ with :

- $Q_{\lambda j,ext} = Q_{\lambda j} \cup \{q_{in,\lambda j}\}$
- $\Sigma_{\lambda j,ext} = \Sigma_{\lambda j} \cup \Sigma'$
- $q_{0,\lambda j,ext} = q_{in,\lambda j}$
- $Q_{m,\lambda j,ext} = Q_{m,\lambda j}$
- $\delta_{\lambda j,ext}$ is defined as follows:
  1. $\forall q \in Q_{\lambda j}$ and $\forall \sigma \in \Sigma_{\lambda j}$, if $(\delta_{\lambda j}(q, \sigma)!)$ then $\delta_{\lambda j,ext}(q, \sigma) = \delta_{\lambda j}(q, \sigma)$.
  2. $\forall q \in Q_{\lambda j}$ from which $\alpha_{\lambda j,\lambda i}$ can occur, then $\delta_{\lambda j,ext}(q, \alpha_{\lambda j,\lambda i}) = q_{in,\lambda j}$.

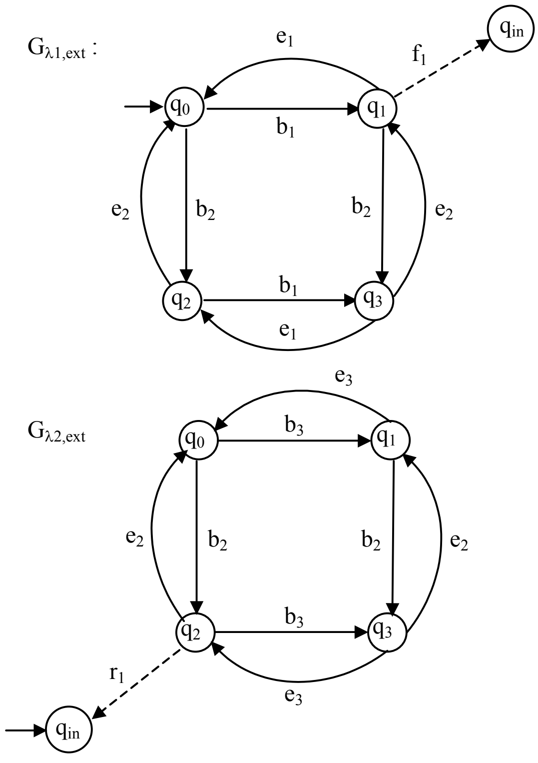


Fig. 3. Extended models of $G_{\lambda 1}$ and $G_{\lambda 2}$

The objective now is to define $\delta_{\lambda j,ext}(q_{in,\lambda j}, \alpha_{\lambda i,\lambda j})$.

[2] $\delta_{\lambda i}(q, \sigma)!$ means that $\delta_{\lambda i}(q, \sigma)$ is defined

Note that initially $G_{\lambda j,ext}$ is in inactive state $q_{in,\lambda j}$. At the occurrence of an event $\alpha_{\lambda i,\lambda j}$, $G_{\lambda j,ext}$ must leave $q_{in,\lambda j}$ in order to reach a state $q \in Q_{\lambda j}$. As shown in fig. 3, at the occurrence of event f_1, $G_{\lambda 2,ext}$, which is in q_{in}, will be lead to q_0, q_1, q_2 or q_3. So $G_{\lambda 2,ext}$ becomes nondeterministic. In order to avoid this nondeterministic situation, we propose the following procedure :

Let $R(G_{\lambda j}, q_{in,\lambda j}, \alpha_{\lambda i,\lambda j})$ be the set of reachable states from $q_{in,\lambda j}$ by the occurrence of $\alpha_{\lambda i,\lambda j}$. To determine this set, we introduce $\pi_{\lambda i,\lambda j} : L_{\alpha_{\lambda i,\lambda j}}(G_{\lambda i}, q_{0,\lambda i})^3 \rightarrow (\Sigma_{\lambda i} \cap \Sigma_{\lambda j})^*$ with :

$\pi_{\lambda i,\lambda j}(\varepsilon) = \varepsilon$ and

$$\pi_{\lambda i,\lambda j}(s\sigma) = \begin{cases} \pi_{\lambda i,\lambda j}(s)\sigma \text{ if } \sigma \in (\Sigma_{\lambda i} \cap \Sigma_{\lambda j}). \\ \pi_{\lambda i,\lambda j}(s) \text{ otherwise.} \end{cases}$$

That is, $\pi_{\lambda i,\lambda j}$ is a projection whose effect on a string $s \in (\Sigma_{\lambda i})^*$ is to erase the elements σ of s that do not belong to $(\Sigma_{\lambda i} \cap \Sigma_{\lambda j})$. $\pi_{\lambda i,\lambda j}(s\sigma)$ allows the identification from $G_{\lambda j}$ of the output states of the intersection elements of $G_{\lambda i}$ when $\alpha_{\lambda i,\lambda j}$ occurs. We achieve the projection definition by defining $(\pi_{\lambda i,\lambda j}(s\sigma))_f$ as the last event of string $s\sigma$ over $\pi_{\lambda i,\lambda j}$.

For example, in the fig. 3, we can determine $\pi_{\lambda 1,\lambda 2}(b_1) = \varepsilon$, $\pi_{\lambda 1,\lambda 2}(b_1 b_2) = b_2$ and $\pi_{\lambda 1,\lambda 2}(b_2 e_2 b_1) = b_2 e_2$ then $(\pi_{\lambda 1,\lambda 2}(s\sigma))_f = (\pi_{\lambda 1,\lambda 2}(b_2 e_2 b_1))_f = e_2$.

Now from the definition of $\pi_{\lambda i,\lambda j}$ two cases are possible : $(\pi_{\lambda i,\lambda j}(s\sigma))_f = \varepsilon$ (case1) or $(\pi_{\lambda i,\lambda j}(s\sigma))_f \neq \varepsilon$ (case 2).

Case 1 :

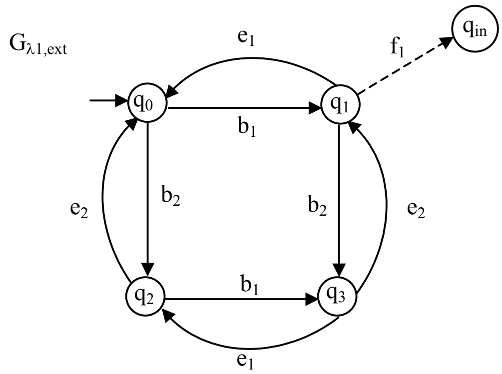
$(\pi_{\lambda i,\lambda j}(s\sigma))_f = \varepsilon$ means that no event of $(\Sigma_{\lambda i} \cap \Sigma_{\lambda j})$ has occurred i.e. no intersection element works.

For example, from fig. 4 we assume that only M_1 is working and $G_{\lambda 1}$ is in q_1 (because of the possible generation of f_1 after the generation of b_1).

Since no event of $\Sigma_{\lambda 2}$ has occurred. So at the occurrence of f_1, $G_{\lambda 2}$ will be lead to the initial state q_0 of $G_{\lambda 2}$.

Thus $R(G_{\lambda 2}, q_{in}, f_1) = q_0$ and $\delta_{\lambda 2,ext}(q_{in}, f_1) = q_0$.

Generally if $(\pi_{\lambda i,\lambda j}(s\sigma))_f = \varepsilon$, then $R(G_{\lambda j}, q_{in,\lambda j}, \alpha_{\lambda i,\lambda j}) = q_{0,\lambda j}$ and so $\delta_{\lambda j,ext}(q_{in,\lambda j}, \alpha_{\lambda i,\lambda j}) = q_{0,\lambda j}$.



[3] $L_{\alpha_{\lambda i,\lambda j}}(G_{\lambda i}, q_{0,\lambda i}) := \{s \in L(G_{\lambda i}) \ / \ post(s) = \alpha_{\lambda i,\lambda j}\}$ with post(s) represents the next event to be occurred after the generation of s.

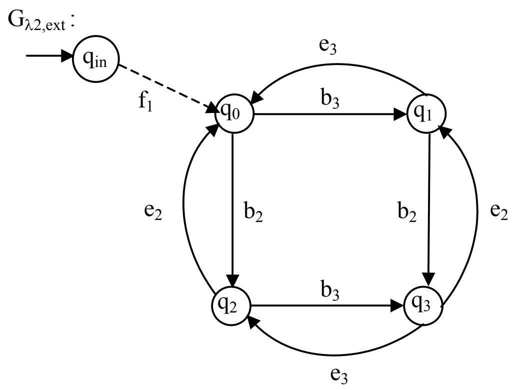


Fig. 4. Extended models of $G_{\lambda 1}$ and $G_{\lambda 2}$ for case 1

Case 2 :

Now suppose that $(\pi_{\lambda i,\lambda j}(s\sigma))_f \neq \varepsilon$ i.e. at least one intersection element is working.

For example, Suppose that in $G_{\lambda 1}$ of fig. 4, b_2 has occurred. Then $(\pi_{\lambda 1,\lambda 2}(b_2))_f = b_2$. So from q_{in}, $G_{\lambda 2}$ can be lead to q_2 or q_3 thus presenting a nondeterministic. To avoid this situation we introduce the following lemma whose proof is not provided here because of the space limitation.

Lemma 1 :

$\delta_{\lambda j,ext}(q_{in,\lambda j}, \alpha_{\lambda i,\lambda j})$ is an unique state which is given by $\delta_{\lambda j,ext}(q_{in,\lambda j}, \alpha_{\lambda i,\lambda j}) = \delta_{\lambda j}(q_{0,\lambda j}, \pi_{\lambda i,\lambda j}(s\sigma))$.

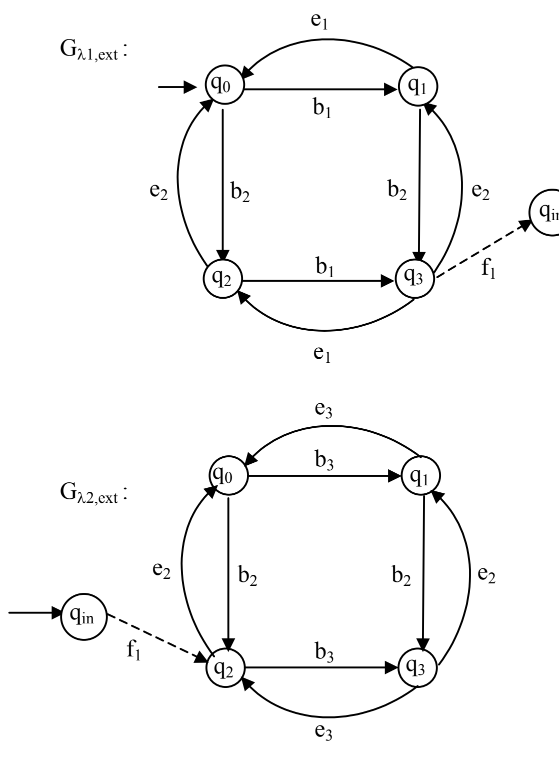


Fig. 5 Extended models of $G_{\lambda 1}$ and $G_{\lambda 2}$ for case 2

From fig. 5 :
$\delta_{\lambda 2,ext}(q_{in}, f_1) = \delta_{\lambda 2}(q_0, \pi_{\lambda 1,\lambda 2}(b_1b_2)) = \delta_{\lambda 2}(q_0, b_2) = q_2$. Thus if $(\pi_{\lambda i,\lambda j}(s\sigma))_f \neq \varepsilon$ then $\delta_{\lambda j,ext}(q_{in,\lambda j}, \alpha_{\lambda i,\lambda j}) = \delta_{\lambda j,ext}(q_{0,\lambda j}, \pi_{\lambda i,\lambda j}(s\sigma))$.

Now suppose that $G_{\lambda i}$ is inactive i.e. $G_{\lambda i}$ is in $q_{in,\lambda i}$ and $G_{\lambda j}$ is active i.e. $G_{\lambda j}$ is in a state $q_{\lambda j} \in Q_{\lambda j}$. If the event $\alpha_{\lambda j,\lambda i}$ occurs, $G_{\lambda j}$ will be inactive but $G_{\lambda i}$ will leave $q_{in,\lambda i}$ to a state $q_{\lambda i} \in Q_{\lambda i}$. We must then as previously define $\delta_{\lambda i,ext}(q_{in,\lambda i}, \alpha_{\lambda j,\lambda i})$. To do this, we introduce $\pi_{\lambda j,\lambda i}$ which is defined as follows:
$\pi_{\lambda j,\lambda i}$: $L_{\alpha_{\lambda j,\lambda i}}(G_{\lambda j}, q_{in,\lambda j}) \rightarrow (\Sigma_{\lambda i}\cap\Sigma_{\lambda j})^*$

$\pi_{\lambda j,\lambda i}(\varepsilon) = \varepsilon$ and

$$\pi_{\lambda j,\lambda i}(s\sigma) = \begin{cases} \pi_{\lambda j,\lambda i}(s)\sigma \text{ if } \sigma \in (\Sigma_{\lambda i}\cap\Sigma_{\lambda j}). \\ \pi_{\lambda j,\lambda i}(s) \text{ otherwise.} \end{cases}$$

For example, from fig. 6 $\pi_{\lambda 2,\lambda 1}(b_3) = \pi_{\lambda 2,\lambda 1}(b_3e_3) = \varepsilon$, but $\pi_{\lambda 2,\lambda 1}(b_3b_2) = b_2$ and $\pi_{\lambda 2,\lambda 1}(b_3b_2e_2) = b_2e_2$.

From the definition of $\pi_{\lambda j,\lambda i}$ two cases, are possible :
$(\pi_{\lambda j,\lambda i}(s\sigma))_f = \varepsilon$ (case 3) or $(\pi_{\lambda j,\lambda i}(s\sigma))_f \neq \varepsilon$ (case 4).

Case 3 :
$(\pi_{\lambda j,\lambda i}(s\sigma))_f = \varepsilon$ means that no event of $\Sigma_{\lambda i}\cap\Sigma_{\lambda j}$ has occurred and $G_{\lambda i}$ must be lead to a state where the intersection elements of $G_{\lambda i}$ and $G_{\lambda j}$ are respectively in their initial states. From fig. 6, q_0 and q_1 are possible in $G_{\lambda 1,ext}$. The objective is to keep only one state by consulting $(\pi_{\lambda i,\lambda j}(s'\sigma'))_f$.

Case 3.a : if $(\pi_{\lambda i,\lambda j}(s'\sigma'))_f = \varepsilon$ then $R(G_{\lambda j}, q_{in,\lambda j}, \alpha_{\lambda j,\lambda i}) = q_{0,\lambda j}$. So from $q_{0,\lambda j}$ and $(\pi_{\lambda j,\lambda i}(s\sigma))_f = \varepsilon$ it can be seen that the intersection elements are in their initial state in $G_{\lambda j}$. Thus when commuting from $G_{\lambda j}$ to $G_{\lambda i}$ we will lead $G_{\lambda i}$ to a state where the intersection elements are in their initial state. This state is inevitably $q_{0,\lambda i}$.
Consequently $\delta_{\lambda i,ext}(q_{in,\lambda i}, \alpha_{\lambda j,\lambda i}) = q_{0,\lambda i}$.

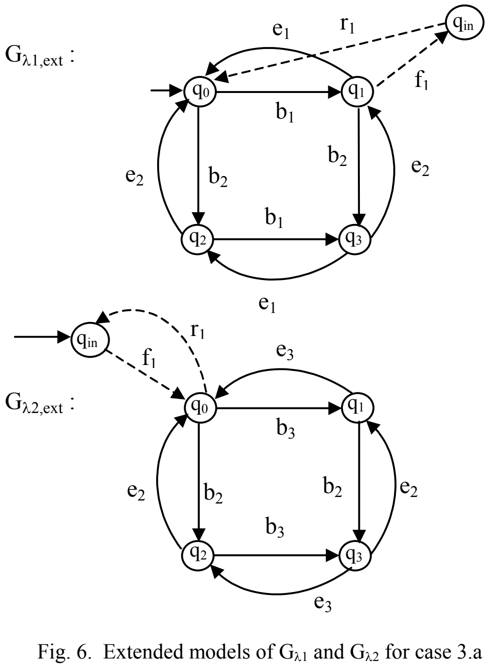


Fig. 6. Extended models of $G_{\lambda 1}$ and $G_{\lambda 2}$ for case 3.a

Case 3.b : if $(\pi_{\lambda i,\lambda j}(s'\sigma'))_f \neq \varepsilon$ then the intersection elements stay at $\delta_{\lambda i}(q_{0,\lambda i}, \pi_{\lambda i,\lambda j}(s'\sigma'))$.
Thus $\delta_{\lambda i,ext}(q_{in,\lambda i}, \alpha_{\lambda j,\lambda i}) = \delta_{\lambda i}(q_{0,\lambda i}, \pi_{\lambda i,\lambda j}(s'\sigma'))$.

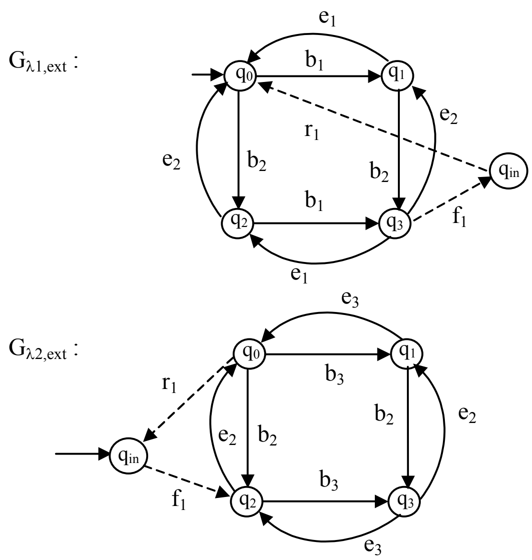


Fig. 7. Extended models of $G_{\lambda 1}$ and $G_{\lambda 2}$ for case 3.b

Case 4 :
Now suppose that $(\pi_{\lambda j,\lambda i}(s\sigma))_f \neq \varepsilon$. At this level two cases can be differentiated : $(\pi_{\lambda i,\lambda j}(s'\sigma'))_f = \varepsilon$ or $(\pi_{\lambda i,\lambda j}(s'\sigma'))_f \neq \varepsilon$.

Case 4.a : $(\pi_{\lambda j,\lambda i}(s\sigma))_f \neq \varepsilon$ and $(\pi_{\lambda i,\lambda j}(s'\sigma'))_f = \varepsilon$. This means that before commutation, no event in $(\Sigma_{\lambda i}\cap\Sigma_{\lambda j})$ has occurred. As shown in case 1, if $(\pi_{\lambda i,\lambda j}(s'\sigma'))_f = \varepsilon$ then
$R(G_{\lambda j}, q_{in,\lambda j}, \alpha_{\lambda i,\lambda j}) = q_{0,\lambda j}$. In the other hand $(\pi_{\lambda j,\lambda i}(s\sigma))_f \neq \varepsilon$ so the events in $(\Sigma_{\lambda i}\cap\Sigma_{\lambda j})$ have occurred from $q_{0,\lambda j}$.
Thus $R(G_{\lambda i}, q_{in,\lambda i}, \alpha_{\lambda j,\lambda i}) = \delta_{\lambda i}(q_{0,\lambda i}, \pi_{\lambda j,\lambda i}(s\sigma))$.
Consequently $\delta_{\lambda i,ext}(q_{in,\lambda i}, \alpha_{\lambda j,\lambda i}) = \delta_{\lambda i}(q_{0,\lambda i}, \pi_{\lambda j,\lambda i}(s\sigma))$.

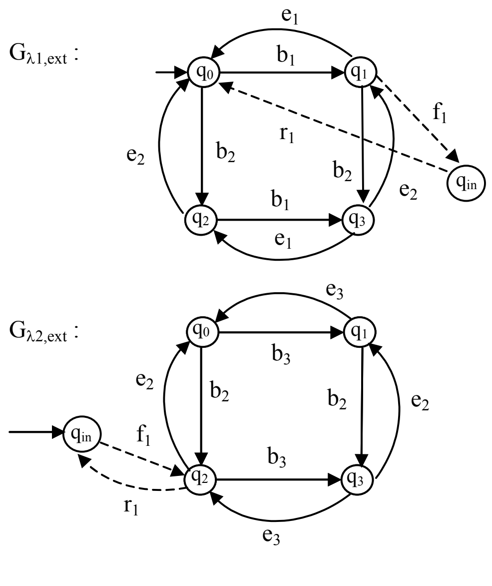


Fig. 8. Extended models of $G_{\lambda 1}$ and $G_{\lambda 2}$ for case 4.a

Case 4.b : $(\pi_{\lambda j,\lambda i}(s\sigma))_f \neq \varepsilon$ and $(\pi_{\lambda i,\lambda j}(s'\sigma'))_f \neq \varepsilon$.
$(\pi_{\lambda i,\lambda j}(s'\sigma'))_f \neq \varepsilon$ lead to $R(G_{\lambda j}, q_{in,\lambda j}, \alpha_{\lambda i,\lambda j}) = \delta_{\lambda j}(q_{0,\lambda j}, \pi_{\lambda i,\lambda j}(s'\sigma'))$ (case 2).
$(\pi_{\lambda j,\lambda i}(s\sigma))_f \neq \varepsilon$ means that events in $(\Sigma_{\lambda i} \cap \Sigma_{\lambda j})$ have occurred from $q_{0,\lambda j}$ i.e. $\delta_{\lambda j}(q_{0,\lambda j}, \pi_{\lambda j,\lambda i}(s\sigma))!$ in $G_{\lambda j}$. We conclude that
$\delta_{\lambda i}(q_{in,\lambda i}, \alpha_{\lambda j,\lambda i}) = \delta_{\lambda i}(q_{0,\lambda i}, \pi_{\lambda i,\lambda j}(s'\sigma')\pi_{\lambda j,\lambda i}(s\sigma))$ i.e.
$R(G_{\lambda i}, q_{in,\lambda i}, \alpha_{\lambda j,\lambda i}) = \delta_{\lambda i}(q_{0,\lambda i}, \pi_{\lambda i,\lambda j}(s'\sigma')\pi_{\lambda j,\lambda i}(s\sigma))$.

From figure 4.b.2, we suppose that $\pi_{\lambda 1,\lambda 2}(s'\sigma') = b_2$ then $\delta_{\lambda 2}(q_0, b_2) = q_2$ in $G_{\lambda 2}$. Now we can introduce the lemma 2 which allows to determine the arrival state when commuting from $G_{\lambda j}$ to $G_{\lambda i}$.

Lemma 2 :
$\delta_{\lambda i,ext}(q_{in,\lambda i}, \alpha_{\lambda j,\lambda i})$ *is an unique state which is given by*
$\delta_{\lambda i,ext}(q_{in,\lambda i}, \alpha_{\lambda j,\lambda i}) = \delta_{\lambda i}(q_{0,\lambda i}, \pi_{\lambda i,\lambda j}(s'\sigma')\pi_{\lambda j,\lambda i}(s\sigma))$.

$(\pi_{\lambda 2,\lambda 1}(s\sigma))_f \neq \varepsilon$ for example $(\pi_{\lambda 2,\lambda 1}(b_3e_2))_f = e_2$ so machine M_2 has finished its task.
Thus $\delta_{\lambda 1,ext}(q_{in}, r_1) = \delta_{\lambda 1}(q_0, \pi_{\lambda 1,\lambda 2}(b_2)\pi_{\lambda 2,\lambda 1}(b_3e_2))$
$= \delta_{\lambda 1}(q_0, b_2e_2) = q_0$ in $G_{\lambda 1}$.

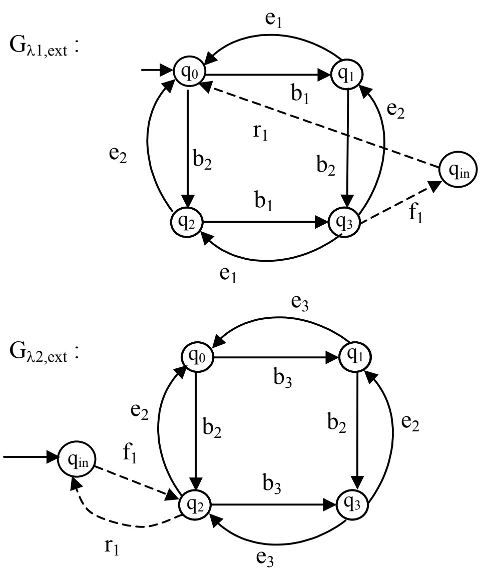


Fig. 9. Extended models of $G_{\lambda 1}$ and $G_{\lambda 2}$ for case 4.b

III. Conclusion

We conclude that the proposed method ensures commutation between different models of a global system reacting to exceptional situations such as a failure event occurrence The major contribution of this paper considers reactive systems with different objectives. Each objective (or operating mode) is represented by a model of the process. Supposing that the different models evolve independently, the main problem is then to inactivate a model $G_{\lambda i}$ and to commute to a model $G_{\lambda j}$ which will be considered as the model of the process until the occurrence of an exceptional event. A formal framework based on tracking events is proposed in order to ensure the commutation. This framework introduces a new projection definition.
Lemma 1 and 2 constitute the main result of this paper. They allow to determine the arrival state of a model after commutation.

References

[1] P. Ramadge and W. Wonham, "Supervisory control of class of discrete event processes", *SIAM Journal of Control and optimisation*, vol. 25, n°1, p. 206-230, 1987.

[2] P. Ramadge and W. Wonham, "Control of discrete event systems", *IEEE transaction on automatic control*, vol. 77, n°1, p. 81-98, January 1989.

[3] P. Ramadge and W. Wonham, "Modular feedback logic for discrete event systems", *SIAM Journal of Control and Optimisation*, vol. 25, n°5, p. 1202-1281, 1987.

[4] F. Lin and W. Wonham, "Decentralised supervisory control of discrete event systems", *Information sciences*, vol. 25, n°5, p. 1202-1218, 1987.

[5] F. Lin and W. Wonham, "On observability of discrete event systems", *Information sciences*, vol. 44, n°2, p. 173-198, 1988.

[6] F. Lin and W. Wonham, "Decentralised control and coordination of discrete-event systems with partial observation", *IEEE transactions on automatic control*, vol. 35, n°12, p. 1330-1337, december 1990.

[7] T. Yoo and S. Lafortune, "New Results on decentralised supervisory control of discrete event systems", *IEEE Conference on Decision and Control 2000*, Sydney, Australia, p. 1-6, december 2000.