Multi-Model Approach for Discrete Event Systems : Application to operating mode management.

O. Kamach, L. Piétrac, 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) where controller result on the supervisory theory control of Ramadge and Wonham [Ram, 87] [Ram, 89]. Our contribution on one hand, enables us to take different views for modeling. In fact each model will represent an operating mode of the process (plant). And on the other hand for each process model we express the associated specifications for the attempted behavior. Our work aim is to manage the operational control of process submitted to failure and the management of operating modes. In this approach, we assume that only one attempted operating mode is active while the others must be put into their respective inactive state. The problem of commutation and tracking between all the designed models is formalised by a proposed framework.

I. INTRODUCTION

Discrete Event Systems (DES) are a special type of dynamic system. The "state" of these systems change at discrete instants in time and the term "event" represents the occurrence of discontinuous change. Different DES models are currently used for specification, verification and synthesis. DES formalism insures the analysis and the assessment of different qualitative and quantitative properties of existing physical systems. Therefore if technological development extends the functionalities of embedded control and their safe functioning, it can steadily increase the complexity of the modeling and synthesis processes.

In fact, DES controls are more and more coupled with technologies whose main objectives are to get the best performances. Having this in mind, the supervisory control theory of Ramadge and Wonham [Ram, 87] [Ram, 89] [Ram, 87] is very helpful. Firstly by proposing the synthesis of controlled dynamic invariant systems by means of feedback and secondly by proving properties such as controllability and non blocking. However, in this theory, the plant result often in a product of a number of the simple components. Thus, the resulted size of the obtained model increases exponentially with the number of components and synthetising a controller becomes a laborious process. But from an operational point of view if this theory gives the best attempted control, it does not directly allow the management of different operating modes. In fact :

- assuming that all the elements which compose the global process are not needed in each operating mode, additional specifications resulting in active/inactive mode must be defined.
- the defined specifications in each model can be conflicting and could lead to the blocking of the system.

Regarding the verification of the classical properties of supervisory theory control (controllability, blocking...), the management of operating modes requires alternating the modes and tracking the evolution of the models. The above reasons (complexity, management of operating mode,...) lead us to develop a multi model approach.

Multi-model approach consists of representing the complex systems by a set of simple models, where each one is a description of the system in a given operating mode, therefore problems such as mode alternation and model tracking must be evoked. By studying mode alternation, we will define the condition where the commutation is allow, the connection between the modes, the model tracking of the process, and how to activate the corresponding specifications. The model tracking of the process localises the states where the commutation events occurs..

The paper is organized as follows : section 2 is devoted to formalizing the problem of commutation between all designed models of the process. In section 3 we will study the mechanism which activates or inactivates the accommodation specifications according to the changes in situations. This mechanisation is based on the tracking of the process models. In section 4 we will present a simple example. Conclusion is expressed in Section 5.

II. COMMUTATION PROCESS

Guaranteed functioning under failure causing downgraded production, yet still allowing continuity of the service, represents the 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. In this section we are interested in modeling these operating modes by applying a multi-model concept which consists of designing a model process for each operating mode.

We define Λ as a set containing indices of all models composing the global system. Card(Λ) represents the

number of models to be designed. The commutation process is shown in figure 1 for card(Λ) = 2. The commutation is investigated as a channel transmiting information that define the starting state (respectively return state) for each process operating in one specifical mode. The commutation will be ensured by defining the projections $\pi_{\lambda_1 \to \lambda_2}$ and $\pi_{\lambda_2 \to \lambda_1}$.

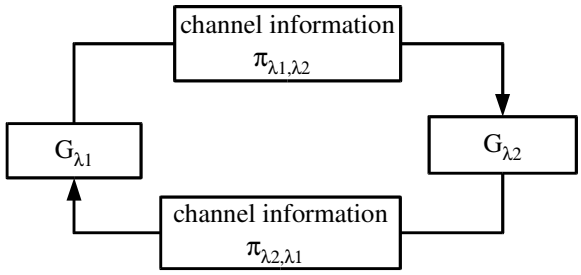


Fig 1: *The information channel in charge of the commutation process.*

Let $\lambda_i \in \Lambda$, {i=1,2}. We define G_{λ_i} as a none controllable DES which is taken to be an automaton of the mode λ_i.
Formally:

$$G_{\lambda_i} = (Q_{\lambda_i},\ \Sigma_{\lambda_i},\ \delta_{\lambda_i},\ q_{0,\lambda_i},\ Q_{m,\lambda_i})$$

We assume that $\Sigma_{\lambda 1} \cap \Sigma_{\lambda 2} \neq \emptyset$ and that initially the process model of the system is G_{λ_1}. All other models like G_{λ_2} stay inactive untill to become activated through $\pi_{\lambda_2 \to \lambda_1}$. Let us consider Σ' the set of the events that are allowed to leave and return to the model G_{λ_1}. At the occurrence of commutation event $\alpha_{\lambda_1,\lambda_2}$ the model of the process becomes G_{λ_2}. However, in this case, we must determine the starting state of G_{λ_2} after commutation. To do this, we first extend G_{λ_1} and G_{λ_2} by adding respectively an inactive state q_{in,λ_1} to the state set of the model G_{λ_1} and an inactive state q_{in,λ_2} to the state set of the model G_{λ_2}. The occurrence of the commutation event $\alpha_{\lambda_1,\lambda_2}$ will lead the model G_{λ_1} to an inactive state q_{in,λ_1} and the model automaton G_{λ_2} will be active from q_{in,λ_2}. So for the model G_{λ_i} the extended model is defined as follows (i = 1, 2):

$$G_{\lambda_i} = (Q_{\lambda_i,ext}, \Sigma_{\lambda_i,ext}, \delta_{\lambda_i,ext}, q_{0,\lambda_i,ext},\ Q_{m,\lambda_i,ext})$$

with :

$$Q_{\lambda_i,ext} = Q_{\lambda_i} \cup q_{in,\lambda_i}$$
$$\Sigma_{\lambda_i,ext} = \Sigma_{\lambda_i} \cup \Sigma'$$
$$q_{0,\lambda_i,ext} = q_{0,\lambda_1} \ \ if\ \ \lambda_i = \lambda_1$$
$$q_{0,\lambda_i,ext} = q_{in,\lambda_i} \ \ if\ \ \lambda_i = \lambda_2$$
$$Q_{m,\lambda_i,ext} = Q_{m,\lambda_i}$$

where the new transition function $\delta_{\lambda_i,ext}$ is defined as follow :

- $\forall$ q $\in$ $Q_{\lambda i}$ and $\forall$ $\sigma \in \Sigma_{\lambda i}$, if $\delta_{\lambda_i}(q,\ \sigma)$ exists, then $\delta_{\lambda_i,ext}(q,\ \sigma) = \delta_{\lambda_i}(q,\ \sigma)$
- $\forall$ q $\in$ $Q_{\lambda i}$ from which $\alpha_{\lambda_i,\lambda_j}$ (i $\neq$ j and i, j $\in$ (1, 2)) can occur, then $\delta_{\lambda_i,ext}(q,\ \alpha_{\lambda_i,\lambda_j}) = q_{\text{in},\lambda_i}$.

The main objective now is the defined the starting state $\delta_{\lambda_2,ext}(q_{in,\lambda_2},\ \alpha_{\lambda_1,\lambda_2})$ and then the return state $\delta_{\lambda_1,ext}(q_{in,\lambda_1},\ \alpha_{\lambda_2,\lambda_1})$.

We note that initially $G_{\lambda_2,ext}$ is in an inactive state q_{in,λ_2} but at the occurrence of the commutation event $\alpha_{\lambda_1,\lambda_2}$, $G_{\lambda_2,ext}$ must leave q_{in,λ_2} in order to reach a state q $\in$ $Q_{\lambda 2}$. Through the information channel, we introduce :

$\pi_{\lambda_1 \to \lambda_2} : \Sigma^*_{\lambda_1} \to \Sigma^*_{\lambda_2}$ such that :

$$\pi_{\lambda_1 \to \lambda_2}(\varepsilon) = \varepsilon$$
$$\pi_{\lambda_1 \to \lambda_2}(s\sigma) = \pi_{\lambda_1 \to \lambda_2}(s)\sigma \ \ if\ \sigma \in \Sigma_{\lambda_1} \cap \Sigma_{\lambda_2}$$
$$\pi_{\lambda_1 \to \lambda_2}(s\sigma) = \pi_{\lambda_1 \to \lambda_2}(s) \ \ if\ \ \sigma \notin \Sigma_{\lambda_1} \cap \Sigma_{\lambda_2}$$

That is, $\pi_{\lambda_1 \to \lambda_2}$ is a projection whose effect on a string s $\in$ Σ_{λ_1} is to erase the element σ of s that does not belong to $\Sigma_{\lambda_1} \cap \Sigma_{\lambda_2}$. The projection $\pi_{\lambda_1 \to \lambda_2}$ identifies from $G_{\lambda 2}$ the output states of the intersection elements of G_{λ_1} when $\alpha_{\lambda_1,\lambda_2}$ occurs. We achieve the projection definition by defining $(\pi_{\lambda_1 \to \lambda_2}(s))_f$ as the last event of string s over $\pi_{\lambda_1 \to \lambda_2}$.

Now from the definition two cases are possible :

$$(\pi_{\lambda_1 \to \lambda_2}(s))_f = \varepsilon \ \ (case\,1)$$
$$(\pi_{\lambda_1 \to \lambda_2}(s))_f \neq \varepsilon \ \ (case\,2)$$

Case 1:

$(\pi_{\lambda_1 \to \lambda_2}(s))_f = \varepsilon$ means no event of $\Sigma_{\lambda_1} \cap \Sigma_{\lambda_2}$ has occurred, namely no intersection element works. So in this case, at the occurrence of the commutation event

$\alpha_{\lambda_1,\lambda_2}$, all the intersection elements G_{λ_1} and G_{λ_2} remain in their initial state.
Generally if $(\pi_{\lambda_1\to\lambda_2}(s))_f = \varepsilon$ then
$\delta_{\lambda_2,ext}(q_{in,\lambda_2}, \alpha_{\lambda_1,\lambda_2}) = q_{0,\lambda_2}$

Case 2:
$(\pi_{\lambda_1\to\lambda_2}(s))_f \neq \varepsilon$ means that at least one common element is working. So the projection
$\pi_{\lambda_1\to\lambda_2}$ provides the information which enables the states of the model G_{λ_1} from which the commutation event $\alpha_{\lambda_1,\lambda_2}$ occurs to be identified.

Proposition 1: *Under the foregoing assumptions* $\forall$ *(s,σ) such that sσ* $\in L(G_{\lambda 1})$ *& follow(sσ)* [1] $= \alpha_{\lambda_1,\lambda_2}$:
$\delta_{\lambda_2,ext}(q_{in,\lambda_2}, \alpha_{\lambda_1,\lambda_2})$ *is a unique state which is given by*
$\delta_{\lambda_2,ext}(q_{in,\lambda_2}, \alpha_{\lambda_1,\lambda_2}) = \delta_{\lambda_2}(q_{0,\lambda_2}, \pi_{\lambda_1\to\lambda_2}(s\sigma))$
❑
Now we suppose that G_{λ_1} is inactive i.e. G_{λ_1} is in an inactive state q_{in,λ_1} and G_{λ_2} is active i.e. G_{λ_2} is in the state q ∈ $Q_{\lambda 2}$. If the event $\alpha_{\lambda_2,\lambda_1}$ occurs, G_{λ_2} will be inactive but G_{λ_1} will leave q_{in,λ_1} to a state q ∈ $Q_{\lambda 1}$. We must then, as previously, define the return state
$\delta_{\lambda_1,ext}(q_{in,\lambda_1}, \alpha_{\lambda_2,\lambda_1})$. We will reciprocally introduce
$\pi_{\lambda_2\to\lambda_1} : \Sigma^*_{\lambda_2} \to \Sigma^*_{\lambda_1}$ with :

$$\pi_{\lambda_2\to\lambda_1}(\varepsilon) = \varepsilon$$
$$\pi_{\lambda_2\to\lambda_1}(s\sigma) = \pi_{\lambda_2\to\lambda_1}(s)\sigma \quad if \quad \sigma \in \Sigma_{\lambda_2} \cap \Sigma_{\lambda_1}$$
$$\pi_{\lambda_2\to\lambda_1}(s\sigma) = \pi_{\lambda_2\to\lambda_1}(s) \quad if \quad \sigma \notin \Sigma_{\lambda_2} \cap \Sigma_{\lambda_1}$$

From the definition of the projection $\pi_{\lambda_2\to\lambda_1}$ two new cases, are possible :
$(\pi_{\lambda_2\to\lambda_1}(s))_f = \varepsilon$ (case 3)
$(\pi_{\lambda_2\to\lambda_1}(s))_f \neq \varepsilon$ (case 4)

Case 3:
$(\pi_{\lambda_1\to\lambda_2}(s))_f = \epsilon$ means that in the model G_{λ_2} no event of $\Sigma_{\lambda_2} \cap \Sigma_{\lambda_1}$ has occurred. With only this information, we cannot identify the state $\delta_{\lambda_1,ext}(q_{in,\lambda_1}, \alpha_{\lambda_2,\lambda_1})$, so we must know the state of the model G_{λ_1} where the commutation event $\alpha_{\lambda_1,\lambda_2}$ has occurred. Two cases are distinguished.

1 follow(sσ) is the following event of the string sσ.

Case 3.a: if $(\pi_{\lambda_1\to\lambda_2}(s'))_f = \varepsilon$ then by *proposition*
$\delta_{\lambda_2,ext}(q_{in,\lambda_2}, \alpha_{\lambda_1,\lambda_2}) = q_{0,\lambda_2}$
So from $q_{0,\lambda 2}$ and $(\pi_{\lambda_1\to\lambda_2}(s))_f = \varepsilon$ it can be seen that the intersection elements are in their initial state in G_{λ_2}. Thus when commuting from G_{λ_2} to G_{λ_1} we will lead G_{λ_1} to a state where the intersection elements are in their initial state. This state is inevitably $q_{0,\lambda 1}$.

Consequently:
$\delta_{\lambda_1,ext}(q_{in,\lambda_1}, \alpha_{\lambda_2,\lambda_1}) = q_{0,\lambda_1}$
Case 3.b:
if $(\pi_{\lambda_1\to\lambda_2}(s'\sigma'))_f \neq \varepsilon$ then the intersection elements stay at $\delta_{\lambda_2,ext}(q_{0,\lambda_2}, \pi_{\lambda_1\to\lambda_2}(s'\sigma'))$. Thus
$\delta_{\lambda_1,ext}(q_{in,\lambda_1}, \pi_{\lambda_1\to\lambda_2}(s\sigma))$
$= \delta_{\lambda_1}(q_{0,\lambda_1}, \pi_{\lambda_2\to\lambda_1}(s'\sigma')\pi_{\lambda_1\to\lambda_2}(s\sigma))$

Case 4:
Now assume that $(\pi_{\lambda_2\to\lambda_1}(s\sigma))_f \neq \varepsilon$. As previously two cases can be distinguished
$(\pi_{\lambda_1\to\lambda_2}(s'\sigma'))_f = \varepsilon$ case (4-a)
$(\pi_{\lambda_1\to\lambda_2}(s'\sigma'))_f \neq \varepsilon$ case (4-b)

Case 4.a: $(\pi_{\lambda_1\to\lambda_2}(s'\sigma'))_f = \varepsilon$**.** This means that before commutation, no event in $\Sigma_{\lambda_2} \cap \Sigma_{\lambda_1}$ has occurred. As shown in case 1, $(\pi_{\lambda_1\to\lambda_2}(s'\sigma'))_f = \varepsilon$ then $\delta_{\lambda_2,ext}(q_{in,\lambda_2}, \alpha_{\lambda_1,\lambda_2}) = q_{0,\lambda_2}$. On the other hand $(\pi_{\lambda_2\to\lambda_1}(s\sigma))_f \neq \varepsilon$ so at least one event in $\Sigma_{\lambda_2} \cap \Sigma_{\lambda_1}$ has occurred from $q_{0,\lambda 2}$. Thus consequently
$\delta_{\lambda_1,ext}(q_{in,\lambda_1}, \alpha_{\lambda_2,\lambda_1})$
$= \delta_{\lambda_1,ext}(q_{0,\lambda_1}, \pi_{\lambda_1\to\lambda_2}(s'\sigma')\pi_{\lambda_2\to\lambda_1}(s\sigma))$

Case 4.b:
$(\pi_{\lambda_2\to\lambda_1}(s\sigma))_f \neq \varepsilon$ and
$(\pi_{\lambda_1\to\lambda_2}(s'\sigma'))_f \neq \varepsilon$
$(\pi_{\lambda_1\to\lambda_2}(s'\sigma'))_f \neq \varepsilon$ lead to

$\delta_{\lambda_2,ext}(q_{in,\lambda_2}, \alpha_{\lambda_1,\lambda_2})$ **(case 2)**.
$= \delta_{\lambda_2}(q_{0,\lambda_2}, \pi_{\lambda_1\to\lambda_2}(s\sigma))$

$(\pi_{\lambda_2\to\lambda_1}(s\sigma))_f \neq \varepsilon$ means that at least one event in $\Sigma_{\lambda_2} \cap \Sigma_{\lambda_1}$ has occurred from $q_{0,\lambda j}$ i.e.
$\delta_{\lambda_1,ext}(q_{in,\lambda_1}, \alpha_{\lambda_2,\lambda_1})$
$= \delta_{\lambda_1}(q_{0,\lambda_1}, \pi_{\lambda_1\to\lambda_2}(s'\sigma')\pi_{\lambda_2\to\lambda_1}(s\sigma))$

Proposition 2 : *$\forall$ (s,s′, σ, σ′) such that sσ ∈ L($G_{\lambda 1}$) & follow(sσ) =* $\alpha_{\lambda_1,\lambda_2}$ *&*
s′σ′ ∈ L($G_{\lambda 2}$, $\delta_{\lambda_2,ext}(q_{in,\lambda_2}, \alpha_{\lambda_1,\lambda_2})$ *)*[2] *& follow(s′σ′) =* $\alpha_{\lambda_1,\lambda_2}$ *then*
$\delta_{\lambda_1,ext}(q_{in,\lambda_1}, \alpha_{\lambda_2,\lambda_1})$ *is a unique state which is given by*

$$\delta_{\lambda_1,ext}(q_{in,\lambda_1}, \alpha_{\lambda_2,\lambda_1}) = \delta_{\lambda_1}(q_{0,\lambda_1}, \pi_{\lambda_1\to\lambda_2}(s'\sigma')\pi_{\lambda_2\to\lambda_1}(s\sigma)) \quad .❑$$

For the notation if $\delta_{\lambda_i}(q, s)$ exists then we note $\delta_{\lambda_i}(q, s)!$.

III. MECHANISM OF COMMUTATION OF SPECIFICATIONS

In this part, we formalise the mechanism which activates or inactivates the corresponding specifications. The reason is that for each operating mode *i.e.* each specification, the resulted control trajectory depends on each starting state. This mechanism is based on tracking the process models and taking into account the active model. For each model G_{λ_i} (for i ≠ j and i, j ∈ {1, 2}) we express the appropriate specification S_{λ_i} for the attemped behavior K_{λ_i}. The problem is that at the occurrence of the commutation event $\alpha_{\lambda_i,\lambda_j}$ the model G_{λ_i} will be inactive, conversely the model G_{λ_j} will be active, so at the same instant we must switch off the specification S_{λ_i} and switch on the specification S_{λ_j}. As previously, the main objective is the determination of the starting state of S_{λ_2} and the return state of S_{λ_1}. The commutation event $\alpha_{\lambda_i,\lambda_j}$ leads the specification S_{λ_i} in the inactive state $x_{in,i}$ but the specification S_{λ_j} must leave the inactive state $x_{in,j}$ to reach a state x in $X_{\lambda j}$.

So for the specification

$$S_{\lambda_i} = (X_{\lambda_i}, \Sigma_{\lambda_i}, \xi_{\lambda_i}, x_{0,\lambda_i}, X_{m,\lambda_i})$$

we give the extended specification as follow : $S_{\lambda_i,ext} = (X_{\lambda_i,ext}, \Sigma_{\lambda_i,ext}, \xi_{\lambda_i,ext}, x_{0,\lambda_i,ext}, X_{m,\lambda_i,ext})$

where :

$$X_{\lambda_i,ext} = X_{\lambda_i} \cup x_{in,\lambda_i}$$
$$\Sigma_{\lambda_i,ext} = \Sigma_{\lambda_i} \cup \Sigma'$$
$$x_{0,\lambda_i,ext} = x_{0,\lambda_1} \ if \ \lambda_i = \lambda_1$$
$$x_{0,\lambda_i,ext} = x_{in,\lambda_i} \ if \ \lambda_i = \lambda_2$$
$$X_{m,\lambda_i,ext} = X_{m,\lambda_i}$$

$\xi_{\lambda_i,ext}$ is defined as follow :

- $\forall$ x ∈ $X_{\lambda i}$ and $\forall$ σ ∈ $\Sigma_{\lambda i}$, if $\xi_{\lambda_i}(x, \sigma)$ exists, then $\xi_{\lambda_i,ext}(x, \sigma) = \xi_{\lambda_i}(x, \sigma)$
- $\forall$ x ∈ $X_{\lambda i}$ from which $\alpha_{\lambda_i,\lambda_j}$ (i # j and i, j ∈ (1,2)) can occur, then

$$\xi_{\lambda_i,ext}(x, \alpha_{\lambda_i,\lambda_j}) = x_{in,\lambda_i}$$

Note that the main objective is the determination of the starting state $\xi_{\lambda_2,ext}(x_{in,\lambda_2}, \alpha_{\lambda_1,\lambda_2})$ and the return state $\xi_{\lambda_1,ext}(x_{in,\lambda_1}, \alpha_{\lambda_2,\lambda_1})$. Let us introduce the projection P_i :

$P_i : (\Sigma_{\lambda_2})^* \to (\Sigma_{\lambda_2})^*$ such that :

$$P_i(\varepsilon) = \varepsilon$$
$$P_i(s\sigma) = P_i(s)\sigma \ if \ \delta_{\lambda_i}(q_i, s\sigma)!$$
$$P_i(s\sigma) = P_i(s) \ if \ \delta_{\lambda_i}(q_i, s\sigma) not!$$

Thus the projection P_i allows the definition of the language of the model G_{λ_2} which takes $q_{\lambda i}$ as the starting state. This language is $P_i(L(G_{\lambda_2}))$

Now the specification

$S_{\lambda_i} = (X_{\lambda_i}, \Sigma_{\lambda_i}, \xi_{\lambda_i}, x_{0,\lambda_i}, X_{m,\lambda_i})$ contains a set of the language $E^k_{\lambda_j}$ [3]

where each language $E^k_{\lambda_j}$ has x_k as the initial state of the specification S_{λ_i}

At the occurrence of the commutation event $\alpha_{\lambda_1,\lambda_2}$, we assume that the starting state of the model G_{λ_2} is $q_{\lambda i}$, from this state the language of the model G_{λ_2} is $P_i(L(G_{\lambda_2}))$.

---

2 L($G_{\lambda 2}$, $\delta_{\lambda_2,ext}(q_{in,\lambda_2}, \alpha_{\lambda_1,\lambda_2})$) = {s ∈ $(\Sigma_{\lambda_2})^*$ / $\delta_{\lambda_2}(\delta_{\lambda_2,ext}(q_{in,\lambda_2}, \alpha_{\lambda_1,\lambda_2}), s)!$

3 $E^k_{\lambda_j}$ ={s ∈ $(\Sigma_{\lambda_j})^*$ / $\xi_{\lambda_j}(x_{\lambda_j}, s)!$ }

Proposition 3: *the starting state*

$\xi_{\lambda_2, ext}(x_{in,\lambda_2}, \alpha_{\lambda_1,\lambda_2})$ *is given by the solution of the following equation : find an unique k such that*

$$P_i(L(G_{\lambda_2})) \cap E^k_{\lambda_2} = K_{\lambda_i}$$

where K_{λ_i} *is the attemped language from* $\boldsymbol{q_{\lambda i}}$ ❑

Now we suppose that the specification S_{λ_2} is active, namely in the state $x_{\lambda 2}$ and S_{λ_1} is inactive i.e. S_{λ_1} is in the inactive state x_{in,λ_1}. If the event $\alpha_{\lambda_2,\lambda_1}$ occurs,

S_{λ_2} will be inactive but S_{λ_1} will leave x_{in,λ_1} to a state $x \in X_{\lambda 1}$. We must then, as previously, define the return state $\xi_{\lambda_1, ext}(x_{in,\lambda_1}, \alpha_{\lambda_2,\lambda_1})$. To this end, we introduce the projection $P_j : (\Sigma_{\lambda_1})^* \rightarrow (\Sigma_{\lambda_1})^*$ where :

$$P_j(\varepsilon) = \varepsilon$$
$$P_j(s\sigma) = P_j(s)\sigma \quad if \quad \delta_{\lambda_j}(q_j, s\sigma)!$$
$$P_j(s\sigma) = P_j(s) \quad if \quad \delta_{\lambda_j}(q_j, s\sigma) not\ !$$

If we assume that the starting state of the model G_{λ_1} is q_j then the language of the model G_{λ_1} is $P_j(L(G_{\lambda_1}))$

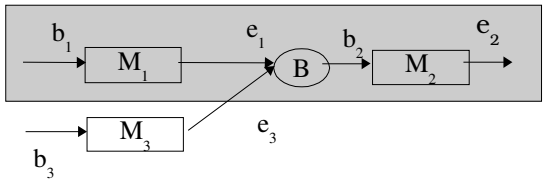
Proposition 4 *: the return state*

$\xi_{\lambda_2, ext}(x_{in,\lambda_1}, \alpha_{\lambda_2,\lambda_1})$ *is given by the solution of the following equation : find an unique k such that*

$P_j(L(G_{\lambda_1})) \cap E^k_{\lambda_1} = K_{\lambda_j}$ *where* K_{λ_j} *is the attemped language from* $\boldsymbol{q_{\lambda j}}$ ❑

IV. ILLUSTRATION

We consider a simple example where the plant is composed of three machines as shown in figure.2 and we only use two operating mode G_n and G_d.
Figure.3.a describes the M_i models and figure.3.b describes two differents global models representing respectively the nominal and degraded mode.
Initially the buffer is empty and M_3 is carrying out another task outside the unit but M_3 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.



bi : beginning of a task on M_i
e_i : end of task on M_i
f_1 : failure of M_1
r_1 : repair of M_1

Fig.2: Scheme of Production unit

We built the extended model of the specification of the nominal mode. For the construction of the extended nominal mode, and degraded mode the reader is referred to [Kam, 02].

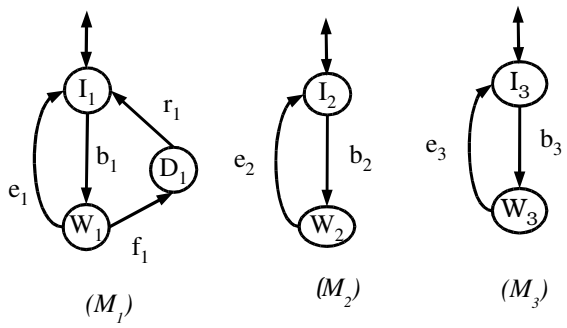


Fig.3.a: Machine models

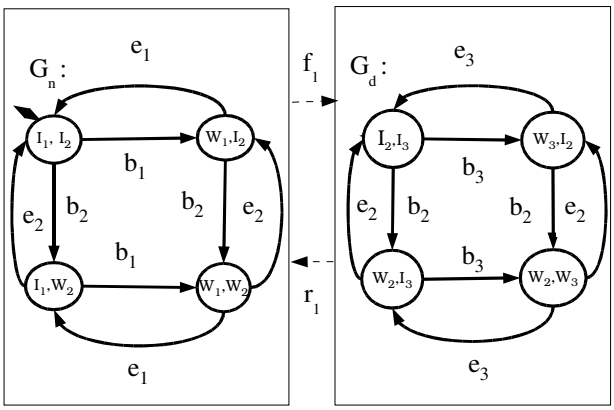


Fig.3.b: The nominal mode and the degraded mode

In this example the set of the commutation events is $\Sigma' = \{f_1, r_1\}$ and the set $\Lambda = \{n, d\}$. As assumed, initially the model of the system is the nominal model G_n, at the occurrence of the commutation event f_1 the model of the system will commute to G_d. This model remains active until the occurrence of the commutation event r_1. The extended models of G_n and G_d are represented in the figure 4.

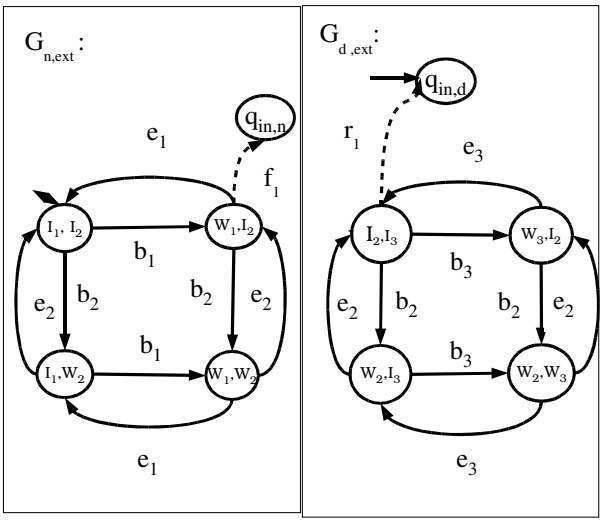


Fig.4: Extended model of G_n and G_d.

The specification of the nominal model is

the buffer must not overflow to 1 or underflow to 0 (see figure. 5).

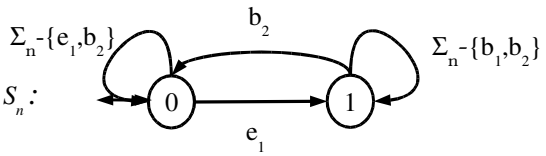


Fig.5: Specification for the nominal mode

The specification of the degraded model is :

- the buffer must not overflow to 1 or underflow to 0 (see figure.6).

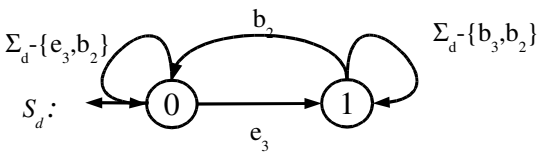


Fig.6: Specification for the degraded mode

Initially the specification of the nominal mode has 0 part at the initial state. However the specification of the degraded model contains two languages

E_d^0 which has 0 part as the initial state and

E_d^1 which has 1 part as the initial state.

At the occurrence of the commutation event f_1 the nominal specification will be switched off but the degraded one will be switched on. Respectively, with the occurrence of r_1 the degraded specification will be switched off and the nominal one will be switched on. The extended nominal specification and the degraded one are represented in figure.7 and 8.

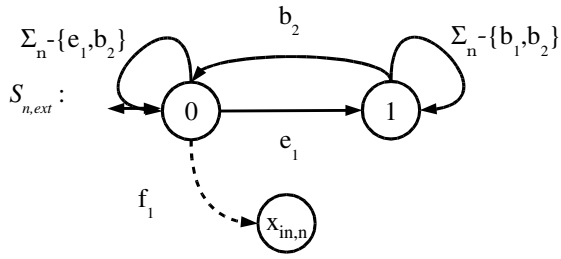


Fig.7: Extended specification of the nominal mode

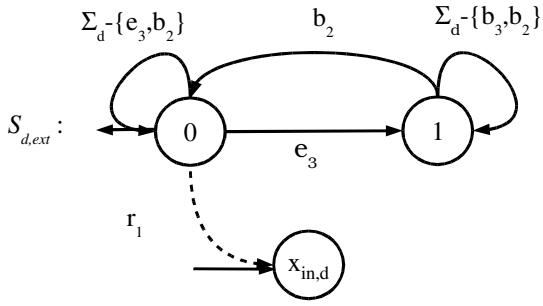


Fig.8: Extended specification of the degraded mode

The attempted language from the state (I_2,I_3) of the degraded mode is

$K_{d_0} = (b_3, b_3 e_3, b_3 e_3 b_2, b_3 e_3 b_2 e_2, \ldots)$. For instance, if we assume that the commutation event f_1 occurred after b_1, then the starting state of the model G_d is (I_2,I_3), therefore the starting state of the degraded specification is :

$P_0(L(G_d)) \cap E_d^k = K_{d_0}$ where $k \in \{0,1\}$. So the k which verified the equation is k =0

The starting state of the degraded specification indicates that no part is available (state 0 as shown in figure 9).

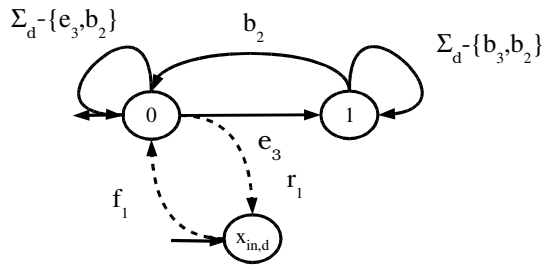


Fig.9: The extended Specification for the degraded mode.

V. CONCLUSION

The proposed approach 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 (*i.e.* operating mode) is represented by a model of the process. Assuming that the different models involve independently, the main problem is then to inactivate the model $G_{\lambda i}$ and to commute to a model $G_{\lambda j}$. $G_{\lambda j}$ 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 definition of the projection function..
Proposition 1, 2, 3 and 4 constitute the main result of this paper. They formally define starting and return state of a model after commutation.

REFERENCES

[Ram, 87] P. Ramadge and W. Wonham, "Supervisory control of class of discrete event processes", ***SIAM Journal of Control and optimisation***, vol. 25, n°1, p. 206230, 1987.

[Ram, 89] P. Ramadge and W. Wonham, "Control of discrete event systems", ***IEEE transaction on automatic control***, vol. 77, n°1, p. 81-98, January 1989.

[Ram, 87] 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.

[Lin, 87] F. Lin and W. Wonham, "Decentralised supervisory control of discrete event systems", *Information sciences*, vol. 25, n°5, p. 1202-1218, 1987.

[Lin, 88] F. Lin and W. Wonham, "On observability of discrete event systems", *Information sciences*, vol. 44, n°2, p. 173-198, 1988.

[Lin, 90] 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.

[Yoo, 00] 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.

[Kam, 02] O. Kamach, S. Chafik, L. Pietrac and E. Niel « representation of a reactive systems with different models », ***IEEE SMC***, Hammamet, Tunisia, reference TA2L4 in CDROM, 6-9 octobre 2002.

[Chafik, 00] S. Chafik "Proposion of an hierarchical-decentralized supervisory control structure : application to the coordination " Ph.D. Thesis, Laboratoire d'Automatique Industrielle, INSA de Lyon, december 2000.