Static approach for switching between different operating modes

A. El Ghadouali, O. Kamach and B. Amami
Laboratoire d'Informatique, Système et Télécommunications, FST de Tanger

***Abstract*—This work deals with operating mode management applied to discrete event systems (DES). Our approach is multi-model based on the information on system states; each model describes a system in a given operating mode. In order to ensure the alternation between these operating modes; we assume that only one attempted operating mode is activated at a time, whilst other modes must be inactivated. The commutation problem may be defined as finding compatible states, when controlled physical system behavior switches from one operating mode to another. For this purpose, we propose an optimal algorithm to find compatibles states when a switching occurred.**

I. INTRODUCTION

In the framework of the safety control of discrete event systems, the abstracted decomposition in modes is a current method in industry to reduce the complexity of a complex system and to describe it. Several works on SED have attempted to help conception of a system through a management of mode [1,2]. However, the operating mode management remains a problem not yet perfectly restrained in the modal decomposition [3,4]. Some studies focus on the use of automata modes to represent modes [5,6] bearing a reasonable size, and others works deal with the problem using several models [7,8]. But, very few take into consideration the problems of commutation between the modes and the validation of alternation mechanism [9, 10, 11].

Different system use corresponds to different operating modes. Adjustment and maintenance modes are examples of other operating modes that are also necessary for the production system. We are interested in modeling these operating modes by applying a multi-model approach [7], which involves designing model process control for each operating mode. We suppose that the system can be engaged only in one operating mode at a time. The commutation between these operating modes takes place when a particular event called commutation events occurs. Using a multi-model approach allows us to materialize very simply the tracking mechanism. The dynamic study of the system is based on succession of generated events. Although, the construction of control by the dynamic approach enables the use of the past events trace, this trace gives much richer information. The determination and the calculation of these traces make the proposed approach more complex and difficult (or impossible) to be applied to real cases. To overcome this problem, we focus on the static approach, by giving priority to states and particular to the current state of the active mode. We will thus present algorithms which are based on the information carried by the state of the system in order to ensure commutation between the operating modes and to find the states compatible when we change the active model. Our approach is based on mathematical approaches and the theory of finite state automata.

This paper is organized as follows:

Section 2 introduces selected DES multi-model design terminology and notation. Section 3 deals with the formalism applicable to the problem of commuting between designed process models. The algorithm allowed to generate the model of faulty mode is also briefly recalled in this section. Section 4 provides a simple example which illustrates the new notions and extended models introduced in this. Study conclusions are presented in Section 5.

II. MULTI-MODEL APPROACH

A system generally presents several operating modes. we propose a multi-model approach that involves representing complex systems by several simple models. Each model characterizes the typology and behavior system involved in these different modes. Activation and inactivation of an operating mode is following the occurrence of a commutation event. This event allows switching from the mode that system performs perfectly its task, known as nominal mode, with a mode for continues a task despite a failure, known as degraded mode. A real system involves a set of nominal and degraded modes. We suppose in our approach the nominal mode will always be considered the first selected mode (initially enabled).

Let a set of operating modes is denoted by $i \in \{1,2,\ldots,n\}$ where $n \in \mathbb{N}$ and $n \geq 2^5$. By convention, we assume initially that the actived mode is mode 1. For each operating mode i, we associate an automata model $G_i=(Q_i, \Sigma_i, \delta_i, q_{i,0}, Q_{i,m})$ where Q_i : is a set of states of mode i, Σ_i : is the alphabet, δ_i: $Q_i \times \Sigma_i \rightarrow Q_i$ is the transition function (partial function), $q_{i,0} \in Q_i$ is the initial state and $Q_{i,m} \subseteq Q_i$ is the subset of marker states. For any state $q \in Q_i$ and event $\sigma \in \Sigma_i$, we write $\delta_i(q,\sigma)!$ (resp. $\delta_i(q,\sigma)\neg!$) if $\delta_i(q,\sigma)$ is defined (resp. isn't defined). The events set Σ of a system is given by the union of all alphabets of elementary automata models increased by the set of commutation events Σ_c. Furthermore, the set of commutation events is disjoint of the different set of models: $\Sigma_c \cap \Sigma_i = \emptyset$ (for $i \in \{1,2,\ldots n\}$). Although, $\Sigma_i \cap \Sigma_j$ can't empty (common components between modes). As for the alphabets of modes, the set of commutation events can be divided into disjoint sets the controllable events and uncontrollable events.

III. COMMUTATION BETWEEN DIFFERENT MODELS

A. *Principle of our approach*

Multi-model approaches to Discrete-Event-Systems (DES) are ideally suited for implementing operating mode management and inter-mode phase switching. Problems such as switching and model tracking must therefore be studied. The operating modes represent the set of acceptable behaviors of the system for a given configuration of resources. Thus, this system doesn't require all components in each operating mode. We assume that the process model can change its structure when commuting from one operating mode to another by engaging new components. For instance, Fig. 1 shows that there are common resources engaged in two operating modes and some resources doesn't contribute to production in mode *i*, but they intervene when a commutation from mode *i* to *j* is performed. Common resource engagements are possible in each considered mode and the concept of tracking is introduced. This means maintaining a trace of events that have occurred for the common components. During this engagement, the structure and the task of the system are fixed. The occurrence of a commutation event leads to the modification of the structure (engagement or disengagement of resources) or the task that will be considered to engage the new system in a different mode.

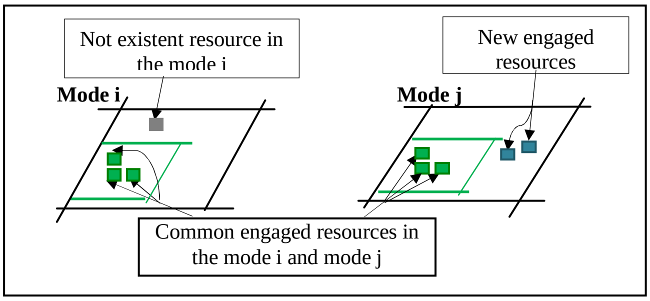

Figure 1. Common resources of the modes *i* and *j*.

The switching mechanism, characterized by information channels, is based on a set of traces generated in the model previously inactivated, to determine a current state for the recently activated model without losing information related to a change of mode. Considering a typology such that there is still a subset of common resources between two modes (see Fig.1), it will be necessary to follow the evolution of this mode to correctly determine the states from which a functional connection will be allowed. The tracking implementation allows determining the connection state, i.e., the state of the adequate novel mode with the state from which the commutation event occurred. This helps us to determine the compatibility when a switching takes place. When a commutation event (failure or repair event) occurs, the system will switch to another operating mode represented by its model G_i. In this case, G_i must be directed to a state compatible with system evolution. The aim is therefore to determine the possible compatible states of each operating mode. Thus, the compatible states are a set of common resources between the two modes that have the same activity in the two states considered.

Our approach aims to ensure a reliable commutation when a mode generates an event enabling the other mode. In our algorithm, we develop mathematical techniques useful to make our approach certain. First of all, we define the notion of the state and compatible state, thereafter; the mathematical elements used in our approach will be detailed.

B. Notations & definitions

Definition 1: (Etat)

A state q_i of the mode *i* is the cartesian product of all activities of resources engaged in this mode.

Example: we suppose that two resources R1 and R2 are involved in the mode *i* respectively with $(a_1^1, a_2^1), (a_1^2, a_2^2)$

A state q_i of mode *i* is written:

$$q_i = (a_l^1, a_k^2) \text{ with } l, k \in \{1,2\} \quad (1)$$

Definition 2: (Compatible state)

A state q_i of the mode is compatible with the state q_j of the mode *j*, where the common resources of these two states have the same activities.

Formally, let E_{ai}, E_{aj} respectively the activities set of the two states q_i and q_j, with:

$\forall a_i^{k\in\mathbb{N}} \in E_{ai}$ and $\forall a_j^{k\in\mathbb{N}} \in E_{aj}$ such that :

$E_{ai}=(a_i^1, a_i^2,..., a_i^n)$ and $E_{aj}=(a_j^1, a_j^2,..., a_j^m)$

Let $R_{ij} = (a_i^l, a_i^{l+1}, \dots, a_i^k)$, with $l \geq 1$ et $k \leq n$, the common resources set between the mode *i* and the mode *j*. The state q_i is compatible with q_j iff

$$E_{ai} \cap E_{aj} \subset R_{ij} \quad (2)$$

Definition 3: (Commutation matrix)

The commutation matrix M_{comut} is a structural representation of the active model $G_a=(Q_a, \Sigma_a, \delta_a, q_{a,0}, Q_{a,m})$. This matrix has the dimension $Na \times Ne$, respectively Na the number of exist activities (lines) and Ne the number of the possible events (columns).

Let M_{comut}: $E_a \times \Sigma_a \rightarrow \mathbb{Z}$ such that

$$M_{comut} = (c_{ij})_{(i,j)\in N^2} = \begin{cases} 1 & \text{if } \exists a \in E_a / \delta_a(a, \sigma_j) = a_i \\ -1 & \text{if } \delta_a(a_i, \sigma_j) \in E_a \\ 0 & \text{if } \delta_a(a_i, \sigma_j) \neg ! \text{(doesn't exist)} \end{cases} \quad (3)$$

With, $\delta_a: E_a \times \Sigma_a \rightarrow E_a$ is the transition function (partial function) in the active mode.

Let $E_{a,i} = (a_i^1, a_i^2, \dots, a_i^n)$ the activities set of a sate q_i in the active mode. For every $a_i^k \in E_{a,i}$ and $\sigma \in \Sigma_a$. We write $\delta_a(a_i^k, \sigma)!$ (resp. $\delta_a(a_i^k, \sigma)\neg!$) if $\delta_a(q_i, \sigma)$ defined (resp. isn't defined). This function can be extended as follows:

$$\delta_a(q_i, \sigma) = \delta_a\left((a_i^1, a_i^2, \dots, a_i^k, \dots, a_i^n), \sigma\right) = \left(\delta_a(a_i^1, \sigma), \delta_a(a_i^2, \sigma), \dots, \delta_a(a_i^k, \sigma), \dots, \delta_a(a_i^n, \sigma)\right) \quad (4)$$

For facilitate the calculating of the compatibility matrix, the transition function δ_a can be written as matrix presented all possible transitions between the activities in the active mode. Thus, we can write:

Let $\delta_a : E_a \times \Sigma_a \rightarrow E_a$ such that

$$\delta_a = (t_{ij})_{(i,j)\in\mathbb{N}^2} \text{ tel que } t_{ij} = a \text{ ssi } \delta_a(a_i, \sigma_j) = a \quad (5)$$

Definition 4: (Occurrence vector)

Let V_o the occurrence vector, it is a resultant vector by the projection of events for a trace of mode *i* in the mode *j*. Its component of order *j* represents the number N how often event σ_j, in the active mode *j*, appears on the activation sequence S_a.

V_o: $\Sigma_j \rightarrow \mathbb{N}$ such that $\forall \sigma \in \Sigma_j$:

$$V_o(\sigma) = \begin{cases} N & \text{if } \sigma \in S_a \\ 0 & \text{else} \end{cases} \quad (6)$$

In order to understand the scope of our proposal, we consider in the first place the case limited to two operating modes. The generation of several modes will be later presented. This generation will not be limited only to increasing the number of modes but will also count for the different possible combinations of switching between modes. Our approach applies to the mechanism for switching between different models, which have been extended to determine their compatible connection states. Finding the states from which these models need to be activated, whilst ensuring adequacy between current process dynamics and control decisions, has solved the problem of the mechanism for switching between models.

Each existing state in the active model is described by set of its activities. First of all, we determine the activities characterized the compatible state. Let S_a an events sequence generated in the inactive model by the arrival of commutation event. The states connected in the activation sequence S_a are selected by the product of the commutation matrix and the number occurrences events of the active mode in this sequence. Taking into account the information related to the initial state allows us to determine the activities of inexistent resources in the inactive mode.

Proposition 2: (Fundamental equation)

The vector activities of compatible state $E_{a,c}$ in the model newly active $G_a=(Q_a, \Sigma_a, \delta_a, q_{a,0}, Q_{a,m})$ when a commutation takes place, it will be determined by:

$$P_a\,(E_{a,c})=P_a\,(E_{a,0})+ M_{comut}\times V_o(S_a) \quad (7)$$

Where, S_a is the sequence of generated events in the model disabled by the occurrence of a commutation event. The vector $E_{a,0}$ is activities of initial state.

Definition 3: (Projection matrix)

The projection P_a presents the projection vector of activities state q_i upon the activities set E_a of existing activities in the active mode. This is defined as follow:

Let $P_a : E_{a,i} \rightarrow \{1,0\}$ such that:

$$P_a\left(E_{a,i}\right) = (p_k)_{k\leq na} = \begin{cases} 1 \ if \ \ \exists a \in E_{a,i}/E_a(k) = a \\ 0 \ \ else \end{cases} \quad (8)$$

The compatible state $q_{a,c}$ is given by the projection of its activities on the descriptive activities matrix $P_{q/a}$ of existing states in the active mode.

$$q_{a,c} = P_{q/a} \times P_a\,(E_{a,c}) \quad (9)$$

The projection matrix $P_{q/a}$ presents projection of each existing state on the activities set in the active mode. This is a matrix of dimension $Nq\times Na$ with Nq is the number of sates $Q=(q_i)_{i\leq nq\in\mathbb{N}}$ of the model (lines) and Na is the number of exist activities $E_i=(a_j)_{j\leq na\in\mathbb{N}}$ in the mode (columns).

Let $P_{q/a} : Q_a \times \mathrm{E}_a \rightarrow \{0,1\}$ such that:

$$P_{q/a} = (p_{ij})_{(i,j)\in\mathbb{N}^2} = \begin{cases} 1 \ \ if \ \exists a \in E_{a,i}/E_a(j) = a \\ 0 \ \ else \end{cases} \quad (10)$$

In order to include in consideration several operating modes, this generalization consists in defining n operating modes and the m possible commutation. The set Σ_c of commutation events is defined as $\cup_{i,j,i\neq j}^{n}\left\{\alpha_{i,j}\right\}$ where $\alpha_{i,j}$ presents the event ensuring the commutation between mode i to mode j. The management modes consist to in ensure the m permitted path of the n modes retained by the designer. Moreover, from a mode i a commutation event can lead to another mode k where i and k are bound by any particular order relation. In other terms, the classification of the modes doesn't present an activation order. Thus the commutation event $\alpha_{i,k}$ (with $i\#k$) occurs, the mode model become G_k model. Then the occurrence of a commutation event $\alpha_{k,j}$ can switch to a model G_j where j can as well be lower or higher than k (with also $j\#k$). This commutation mechanism is illustrated by Fig.2.

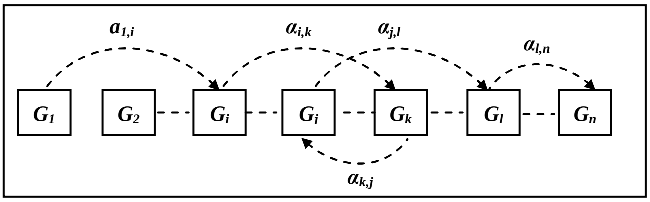

Figure 2. Commutation between different model modes

Proposition 3: (Fundamental equation)

Considering several operating modes, the compatible state is defined as its vector activities $E_{a,c}$ in the model newly active $G_a=(Q_a, \Sigma_a, \delta_a, q_{a,0}, Q_{a,m})$ when activation sequence S_a takes place, it is given as follow:

$$P_a\,(E_{a,c})=P_a\,(E_{a,0})+ M_{comut}\times V_o(S_a) \quad (11)$$

Where, The vector E_0 is given by the horizontal concatenation of all vectors activities of initial state in each mode: $E_0=\{ E_{1,0},\ldots, E_{i,0},\ldots, E_{j,0},\ldots, E_{a,0}\}$

The commutation matrix is defined as the direct sum of the commutation matrix characterized in each mode:

$M_{comut}=\{ M_{comut,\,1}\oplus\ldots\oplus M_{comut,\,i}\oplus\ldots\oplus M_{comut,\,j}\oplus\ldots\oplus M_{comut\,,\,a}\}$

The occurrence vector is given by the vertical concatenation of all projection of events for trace in each mode: $V_o(S)=\{ V_{o1}(S);\ldots;V_{oi}(S);\ldots;V_{oj}(S);\ldots;V_{oa}(S)\}$

All these history compatibility are required to determine the starting states in each mode to which switching leads. Based on the definition of the mathematical elements, we expose thereafter the frame of the proposed algorithm for determine the compatible states in an active mode.

C. Algorithm

The algorithm proposed, gives a fundamental equation to find the compatible states when the system switches from an operating mode to another mode. This algorithm was based on the concepts defined in advance. It is about the transition matrix, the projection matrix, the commutation matrix and the vector of occurrence. Our algorithm receives as input a finite state automaton $G_a=(Q_a, \Sigma_a, \delta_a, q_{a,0}, Q_{a,m})$ representing the global model of the active mode, and sequence of activation. The latter is a sequence one of the possible traces in the model previously inactivated arriving at a commutation event. This algorithm has only one output which presents the projection vector of state compatible. The proposed algorithm is presented as follows:

```
Algorithm: Research of the compatible states
Input:
Pa[] : array [1, nq ; 1, na] of characters;
σa [] : array [1, ne] of characters;
Ea [] : array [1, na] of characters;
Ma[ ; ] : array [1, na ; 1, ne] of characters;
Sa[] : array [1, ns] of characters;
qa,0[] : array [1, nq] of integer;
Output:   qa,c[] : array [1,nq]of integer;
Variables :
i,j,k,l,h : integer
Mcomut[ ; ] : tableau [1, na ; 1, ne] of integer;
Vo [] :array [1, ne] of integer;
M [] : array [1, na] of integer;
Ea,c [] : array [1, na] of integer
Begin
{Determinate the occurrence vector}
for  j : 1 to ne do
    for l : 1 à ns faire
       if Sa [l]==σa [j]  than Vo[j]:= Vo[j]+1 ;
       endif
     end
 end
 {Determinate the commutation matrix}
 Ea,0=(Pa)^t x qa,0;    { ( )^t transpose of a matrix }
 for i :1 to na do
     for j :1 to ne do
      k =1; Mcomut [i j]==0;
        while (k<= na and Mcomut [i j] == 0)  do
          if δa [i j]= Ea [k]        than Mcomut [i j]:=-1;
          else if δa[k j] = Ea [i] than Mcomut [i j]:=1;
                                            else Mcomut[i j]:=0;
             k :=k+1;
        endif
           endif
       end
 M [i] = M [i] + Mcomut [i j]* Vo[j];
    end
 Ea,c [i] = Ea,0[i] + M [i];
 end
{Determinate the compatible state}
while (h <= na and Pa [h, :] !=(Ea,c)^t ) do h :=h+1 ;
end
qa,c:=q(h);
end
```

IV. Application

To facilitate the understanding of the proposed approach, we treat as an example a production system illustrated in Fig.3. This system is composed by four different machines M_1, M_2, M_3 and M_4, and by one buffer S. The machines are used to process a part and the buffer is used as storage between the components with a maximal capacity of 1.

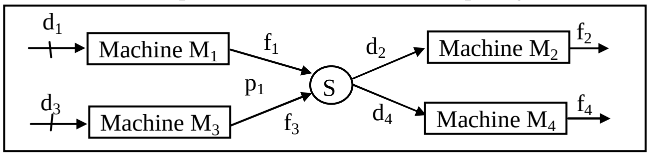

Figure 3. Production system with four machines and intermediate stock.

In this system, the machines operate independently, each machine M_i picks up a work piece from an infinite bin (modeled by the controllable event d_i) and places (b) it in buffer after completing its work (symbolized by the occurrence of the uncontrollable event f_i). The machine M_1 can break down due to malfunctioning and this fact is modeled using the event p_1. Repair is modeling using the event r_1. These switch events involve a switch for the system from nominal to degraded mode. The system has two functioning modes. The first one is a nominal mode where only the components M_1, M_2 and M_4 are used. However, the machine M_1 may fail. It is the degraded mode. In this mode, the machine M_1 is replaced by the machine M_3. In this example, it is assumed that only M_1 can break down (event p_1) and be repaired (event r_1).

The machine M_i are modeled by the automaton denoted G_i and are shown Fig.4.(a) for the machine M_1 and Fig.4.(b) for the components M_2, M_3 and M_4. The dotted arrows represent the commutation events.

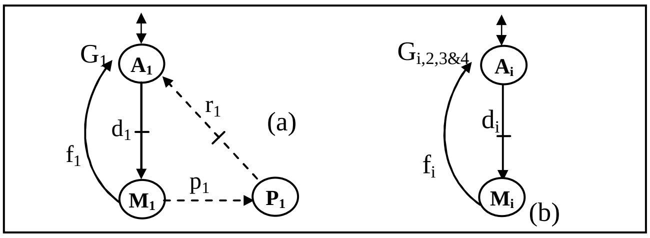

Figure 4. (a) Automata model G_1 of machine M_1.(b) Automata models G_i of machines 2, 3 et 4.

Initially, the system runs in the nominal mode described by the model G_n shown Fig.5. When f_1 occurs, the system switches to the degraded mode described by the model G_d shown Fig.6. Occurrence of r_1 allows G_d to switch to G_n. This means that only one operating mode is activated at one time. The commutation events don't appear in these models. Indeed, these events will be considered for the passage from a mode to the other.

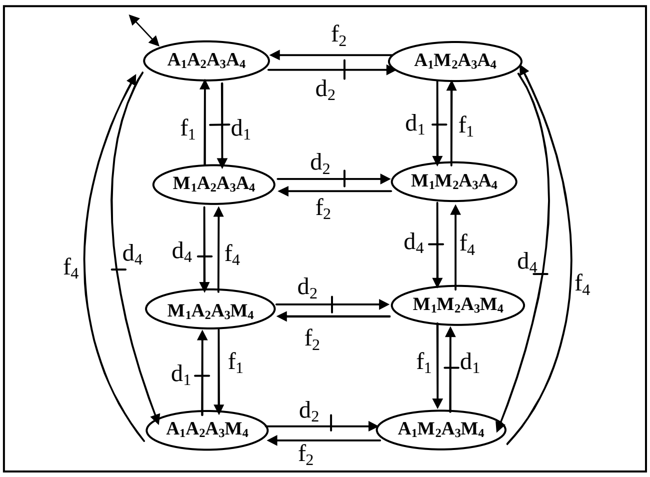

Figure 5. Automata model of nominal mode

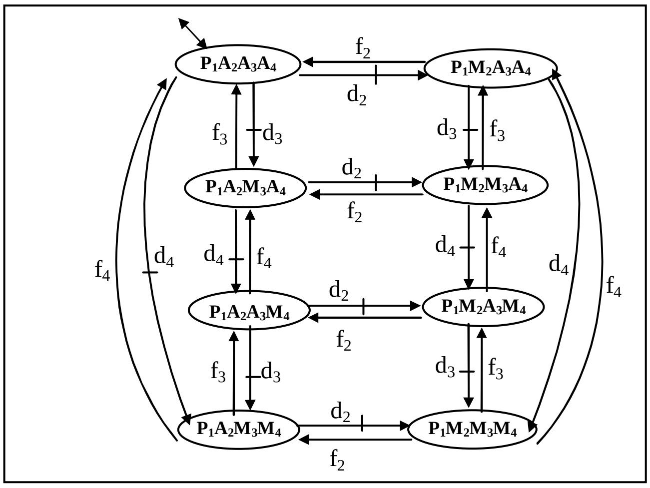

Figure 6. Automata model of degraded mode.

We apply the different steps proposed in the models of two automata nominal and degraded modes in the presence of commutation events $\{p_1,r_1\}$ when an alternation takes place. We consider the active mode, after the occurrence of the failure event p_1, is the degraded mode G_d. The element characterized (input of our algorithm) of the active mode (degraded mode G_d) are given as follow:

- The states set of degraded mode:

$$Q_d=\begin{pmatrix} \text{'P}_1\text{A}_2\text{A}_3\text{A}_4\text{','P}_1\text{M}_2\text{A}_3\text{A}_4\text{','P}_1\text{A}_2\text{M}_3\text{A}_4\text{','P}_1\text{M}_2\text{M}_3\text{A}_4\text{',} \\ \text{'P}_1\text{A}_2\text{A}_3\text{M}_4\text{','P}_1\text{M}_2\text{A}_3\text{M}_4\text{','P}_1\text{A}_2\text{M}_3\text{M}_4\text{','P}_1\text{M}_2\text{M}_3\text{M}_4\text{'} \end{pmatrix}$$

- The events set of degraded mode:

$$\Sigma_d=\{\, d_2, d_3, d_4, f_2, f_3, f_4\}$$

- The activities set of degraded mode:

$$E_a=\{\, a_2, a_3, a_4, m_2, m_3, m_4\}$$

- The transition matrix of degraded mode (5):

$$\delta_a=\begin{bmatrix} a_2\\ a_3\\ a_4\\ m_2\\ m_3\\ m_4\end{bmatrix}\overset{\begin{bmatrix} d_2 & d_3 & d_4 & f_2 & f_3 & f_4\end{bmatrix}}{\begin{bmatrix} m_2 & 0 & 0 & 0 & 0 & 0\\ 0 & m_3 & 0 & 0 & 0 & 0\\ 0 & 0 & m_4 & 0 & 0 & 0\\ 0 & 0 & 0 & a_2 & 0 & 0\\ 0 & 0 & 0 & 0 & a_3 & 0\\ 0 & 0 & 0 & 0 & 0 & a_4\end{bmatrix}}$$

- The projection matrix P_a of degraded mode (8):

$$P_a=\begin{bmatrix} P_1A_2A_3A_4\\ P_1M_2A_3A_4\\ P_1A_2M_3A_4\\ P_1M_2M_3A_4\\ P_1A_2A_3M_4\\ P_1M_2A_3M_4\\ P_1A_2M_3M_4\\ P_1M_2M_3M_4\end{bmatrix}\overset{[\, a_2 \;\; a_3 \;\; a_4 \;\; m_2 \;\; m_3 \;\; m_4\,]}{\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0\\ 0 & 1 & 1 & 1 & 0 & 0\\ 1 & 0 & 1 & 1 & 0 & 0\\ 0 & 0 & 1 & 1 & 1 & 0\\ 1 & 1 & 0 & 0 & 0 & 1\\ 0 & 1 & 0 & 1 & 0 & 1\\ 1 & 0 & 0 & 0 & 1 & 1\\ 0 & 0 & 0 & 1 & 1 & 1\end{bmatrix}}$$

Based on the proposed algorithm, the commutation matrix (3) of our system is:

$$M_{comut}=\begin{bmatrix} a_2\\ a_3\\ a_4\\ m_2\\ m_3\\ m_4\end{bmatrix}\overset{\begin{bmatrix} d_2 & d_3 & d_4 & f_2 & f_3 & f_4\end{bmatrix}}{\begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0\\ 0 & -1 & 0 & 0 & 1 & 0\\ 0 & 0 & -1 & 0 & 0 & 1\\ 1 & 0 & 0 & -1 & 0 & 0\\ 0 & 1 & 0 & 0 & -1 & 0\\ 0 & 0 & 1 & 0 & 0 & -1\end{bmatrix}}$$

Let $S_a=d_1\ d_2\ f_2\ d_4\ d_2$ an activation sequence of degraded mode G_d. The latter is a sequence of the possible traces in the set Σ_n^* of the nominal mode arriving at a commutation event p_1. We determine the occurrence vector (6) by applying the activation sequence S_a. Thus, we obtain: V_o=(2,0,1,1,0,0).

By applying the equation fundamental (7), the compatible state of the degraded model G_d, when commutation event p_1 occurs, is given as follow:

$$P_a\,(E_{d,c})= P_a\,(E_{d,0})+ M_{comut} \times V_o(S_a)$$

Where, S_a is the sequence of events generated in the nominal model disabled by the occurrence of a commutation event p_1. And $E_{d,0}$ is the vector activities of initial states in the degraded mode: $E_{d,0}=\{a_2,a_3,a_4\}$ = (1;1;1;0;0;0)

$$P_a\,(E_{d,c})=\begin{bmatrix}1\\1\\1\\0\\0\\0\end{bmatrix}+\begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0\\ 0 & -1 & 0 & 0 & 1 & 0\\ 0 & 0 & -1 & 0 & 0 & 1\\ 1 & 0 & 0 & -1 & 0 & 0\\ 0 & 1 & 0 & 0 & -1 & 0\\ 0 & 0 & 1 & 0 & 0 & -1\end{bmatrix}\times\begin{bmatrix}2\\0\\1\\1\\0\\0\end{bmatrix}=\begin{bmatrix}0\\1\\0\\1\\0\\1\end{bmatrix}$$

Thus, $P_a(E_{d,c})= P_a(a_3, m_2, m_4)$= "$P_1M_2A_3M_4$"

The resulting vector $q_{d,c}$ corresponds to the activation of the initial state in the degraded model G_d and the inactivation of other states in the same model. So, the attainable state in the degraded model G_d after the sequence generated $S_a=d_1\ d_2\ f_2\ d_4\ d_2$, is "$P_1M_2A_3M_4$". This state is compatible with the nominal mode because the activity of the common resource M_2 and M_4 is found in the state "$M_1M_2\ A_3M_4$" after the occurrence of the sequence S_a and before the occurrence of the commutation event p_1. Therefore, our algorithm ensures the identification of compatible states.

We apply our algorithm on all the traces of the nominal mode immediately before the failure event and the degraded mode tracking by the repair event, we obtain the model in Fig. 7.

V. CONCLUSION

This paper deals with the compatibility of states when a switching between the operating modes takes place. It introduces a definition of compatible states and gives a solution to resolve the management alternation problem of different operating modes. The compatibility was considered as a current state when a mode generates an event activated the other mode. We proposed an algorithm able to recognize these states with low complexity and based on a static approach. Our current research is attempting to resolve the compatibility problem in real time by using the timed automata as a modeling tool.

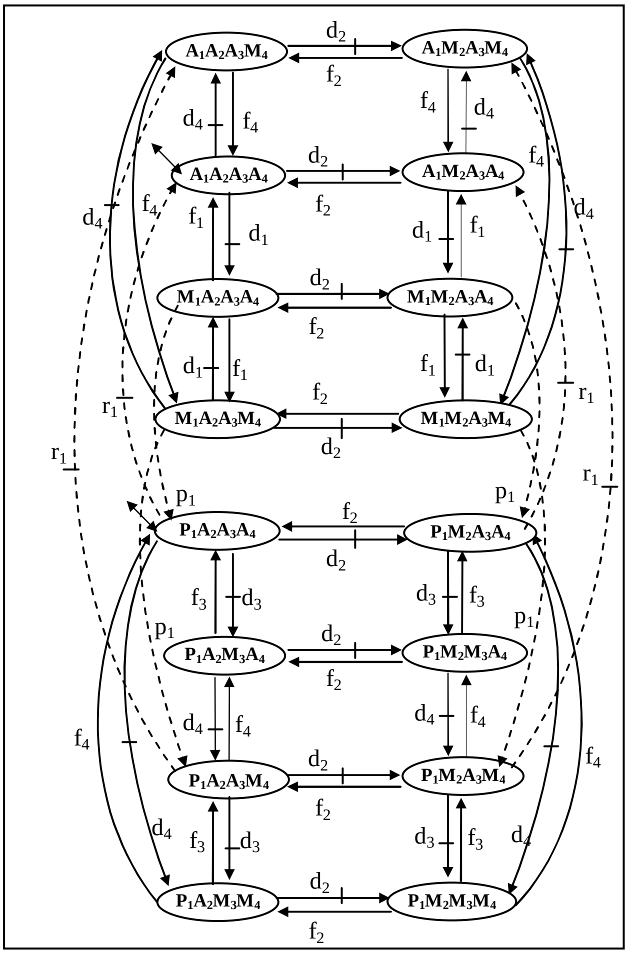

Figure 7. Commutation between nominal mode and degraded mode.

REFERENCES

[1] N. Dangoumau ,A. Toguyéni, E. Craye , “Functional and behavioural modelling for dependability in automated production systems”, *Proceedings of the Institution of Mechanical Engineers, Part B : Journal of Engineering Manufacture*, vol. 216, n° 3,2002, p. 389-405.

[2] M. Zefran, and J. Burdick, “Design of switching controllers for systems with changing dynamies,” *37th Conference on Decision and control(CDC)*, vol. 2, 1998,p. 2113 – 2118.

[3] M. Nourelfath, and E. Niel, “Modular supervisory control of an experimental automated manufacturing system,” *Control Engineering Practice-Journal of IFAC*, vol. 12, n°2, 2004,p. 205–216.

[4] P. Charbonnaud, F. Rotellaand, and S. Màuoar, “Process operating mode monitoring process : switching online the right controller,”*IEEE Transactions on Control System Technology*, vol.31, 2002.

[5] F. Maraninchi and Y. Rémond. “Mode-automata : a new domain-specific construct for the development of safe critical systems,”*Science of Computer Programming*, vol.1, n°16, 2003, p. 219–254.

[6] J-P. Talpin, C. Brunette, T. Gautier and A.Gamatie, “Polychronous mode automata,” *In EMSOFT ’06 : Proceedings of the 6th ACM & IEEE International conference on Embedded software*, New York, NY, USA, 2006, p.83–92.

[7] O. Kamach, L. Piétrac and E. Niel, “Multi-model approach to discrete events systems: application to operating mode management,”*Mathematics and Computers in Simulation*, vol. 70, n°5-6, 2005, p. 394-407.

[8] Faraut G., Piétrac L., Niel E., « Formal Approach to Multimodal Control Design : Application to Mode Switching », IEEE Transactions on Industrial Informatics, vol. 5, n° 4, 2009, p.443-453.

[9] E. Asarin, O. Bournez, Drang, O. Maler and A.Pnueli, “Effective synthesis of switching controllers for linear system,”*Proceedings of IEEE*,vol. 88, n°7, 2000, p. 1011-1025.

[10] O. Kamach, L. Piétrac and E. Niel, “Repulsive/ Attractive discrete state space sets for switching management,”Studies in Informatics and Control Journal, vol.16, n°1, 2007, pp. 83-96.

[11] G. Faraut, L. Pi´etrac, and E. Niel, “Identification of incompatible statesin mode switching”, *Emerging Technologies and Factory Automation,2008. ETFA 2008.IEEE International Conference on*,2008, pp. 121–128.