Repulsive/Attractive Discrete State Space Sets for Switching Management

Oulaïd KAMACH, Éric NIEL and Laurent PIÉTRAC
INSA de Lyon, Laboratoire d'Automatique Industrielle (LAI)
Bat. Antoine de St-Exupéry
25, Av. Jean Capelle
http://www-lai.insa-lyon.fr
69621 Villeurbanne CEDEX France
laurent.pietrac@insa-lyon.fr

Abstract

This paper deals with operating mode management of Discrete Event Systems (DES) and this contribution is based on Supervisory Control Theory (SCT). Our aim is to extend SCT by introducing a mechanism for managing different operating modes for the controlled system. An operating mode corresponds to a specific system structure (engagement or disengagement of different system components) and specified tasks. Mode management will consist in controlling switching between modes with a view to handling models of reasonable size. Our approach is a multi-model one and involves representing a complex system by a set of simple models, each of which describes the system in a given operating mode. The adopted approach assumes that only one attempted operating mode is activated at a time, whilst other modes must be deactivated. The switching problem may be defined as finding compatible states, when controlled system behavior switches from one operating mode to another. The major contribution of this paper is the avoidance of switching from states (called forbidden states) with ghost compatible states in the selected operating mode. These states are called ghost because their existence would potentially violate a defined selected mode specification.

1 Introduction

Operating mode management for DES remains a challenging problem and is the subject of considerable research [1, 16, 4, 5, 2, 11]. Existing work on operating mode management for DES focuses on problems of characterisation and switching between modes [1, 2]. However, these approaches are not based on any formal models: they possess neither any validation mechanism of possible alternations (enabling and validity of switching between modes) nor any validation mechanism of deadlock research. To overcome these drawbacks in the Dynamic Hybrid Systems context, most works suggest novel methodology for synthesizing switching controllers and define the synthesis problem as finding the condition on which a controller should switch system behavior from one mode to another to avoid a set of bad states [4]. [16] presents a framework for designing stable control schemes for systems, whose dynamic equations change as they evolve in several operating modes. An appealing alternative is switching control schemes. Here, a different controller is applied to each operating mode and the stability of the overall system is ensured through a suitable switching scheme. In approach of [5], a supervised control structure integrating operating mode detection and an active accommodation loop is designed.

Active control accommodation is based on indirect switching control because it depends on detection of the actual process model.

Based on SCT (initiated by Ramadge and Wonham [12]), the approaches proposed by [11] and [3] apply the macro-action concept; operating mode management is ensured by activation of only one mode at any one time. Conscious of the advantages offered by [11] and [3], we extend these approaches to take the following statements into account:

1. A process comprises several components and not all components are used in every operating mode.
2. Specifications defined for each model can be conflicting, when switching from one mode to another (unlike the approach [6] in which all objectives must be concurrently achieved) and this may cause system blocking.

We have introduced a framework for modeling and switching, which takes into account the above statements [7]. The models considered feature processes and specifications, and more specifically, components engaged in a given operating mode. The multi-model approach involves representing a complex system by several simple models (each process model is associated with a specification model in a given operating mode). Each model is a partial description of the system in a given operating mode. Initially, only one model is activated and the nominal operating mode is generally assumed. All other modes are deactivated. Common component 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. We have therefore extended each considered process and specification model by adding a specific state called the inactive state. The set of the events making it possible to switch from one model (process and specification) to another is called the set of the switching events. The difficulty of such an approach resides both in the building of extended models, which characterise different operating modes and in defining a switching mechanism allowing us to track explicitly the behavior of each model. This switching mechanism, characterized by information channels, is based on a set of traces generated in the model previously deactivated, to determine a suitable starting or recovery state for the recently activated model. Our approach applies to the mechanism for switching between different process and specification 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 specification models. In this paper, we extend the approach of [7], [8] by considering a problem of switching from states with potentially ghost compatible connection states in the selected operating mode.

In Section 2, switching between modes is ensured by tracking model S_i / G_i to ensure compatibility between the current state and all previous mode changes. Intuitively, a state q in a model S_i / G_i is said to be compatible with a state q' in a model S_j / G_j, if the set of the common components between the two modes i and j have the same activity in the two considered states and the controlled process behavior S_i / G_i (resp. S_j / G_j) corresponds to a defined desired language of mode i (resp. mode j).

Based on Kumar's algorithm [9], we thus develop an algorithm which allows forbidden and pre-forbidden states to be avoided. Proposed definition of multiple forbidden or pre-forbidden starting states in operating mode management allows implementation of more significant switching laws. Section 3 describes a set of transient and terminal modes, which depend on the production rate of the implemented components. Starting state definition prompts switching law management in relation to space set attractiveness/repulsiveness.

2 Multi-model approach

2.1 Principle

A real system involves a set of nominal and degraded modes. We adopt the following notation to deal with this. The set of operating modes is denoted by $I = \{1,2,...,n\}$, where $n \in N$ and $n \geq 2$. By convention, we assume initially that the activated mode is mode 1. For each operating mode i, we associate an automaton model $G_i = (Q_i, \Sigma_i, \delta_i, q_{i,0}, Q_{i,m})$ (Q_i is a set of states of mode i, Σ_i the alphabet, δ_i the transition partial function, $q_{i,0} \in Q_i$ the initial state, $Q_{i,m} \subseteq Q_i$ the subset of marked states) coupled by its proper supervisor S_i. A specification model E_i is also an automaton, $E_i = (X_i, \Sigma_i, x_{i,0}, \xi_i, X_{i,m})$, and the controlled system S_i / G_i is obtained by composition of G_i and E_i, *i.e.* $S_i / G_i := E_i \times G_i = (X_i \times Q_i, \Sigma_i, (x_{i,0}, q_{i,0}), \xi_i \times \delta_i, X_i \times Q_i)$ where $\xi_i \times \delta_i : X_i \times Q_i \times \Sigma_i \rightarrow X_i \times Q_i : (x, q, \sigma) \rightarrow (\xi_i(x, \sigma), \delta_i(q, \sigma))$ provided $\xi_i(x, \sigma)$ exists and $\delta_i(q, \sigma)$ exists. The set Σ' of switching events is defined as $\bigcup_{i,j,i\neq j}^{n} \alpha_{i,j}$, where $\alpha_{i,j}$ represents the event ensuring switching from mode i to mode j. These multiple switching events mean that several switchings are possible: switching from mode 1 to mode 2, switching from mode 1 to i, from mode 2 to k, etc. These switchings must induce a trace memorization step because of common component engagement. Let us consider a case in which switching takes place, from mode i to mode j, then from mode j to mode k. In this case, we have to memorize controlled process S_i / G_i history in mode i prior to initial switching, then controlled process S_j / G_j history in mode j prior to the second switching. All these history recordings are required to determine the starting states in each mode (*i.e.* in each state of process G_i and specification E_i engaged in that mode) to which switching leads. These recordings are performed by the information channel denoted by $\pi_{i,j}$ (Figure 1), where:

Definition 1

Let $\pi_{i,j} : \Sigma_i^* \rightarrow \Sigma_j^*$ *such that* $\forall \sigma \in \Sigma_i$ *and* $\forall \sigma \in \Sigma_i^*$ *:*

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

$$\pi_{i,j}(\varepsilon) = \begin{cases} \pi_{i,j}(s)\sigma & \text{if } \sigma \in \Sigma_i \cap \Sigma_j \\ \pi_{i,j}(s) & \text{if } \sigma \in \Sigma_i - \Sigma_j \end{cases}$$

This projection function definition restricts neither alphabet Σ_i nor alphabet Σ_j. In the particular case in which $\Sigma_j \subseteq \Sigma_i$, this function corresponds to the canonical projection used conventionally in SCT [10, 14, 15]. This

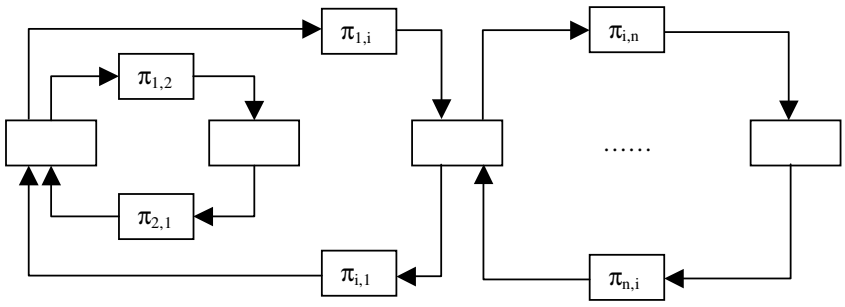

Figure 1: Exchanges of necessary information for modes management

function "erases" effectively from a string *s* those events σ that are not included in the set of common events $\Sigma_i \cap \Sigma_j$. This allows the behavior of common components only to be tracked.

In S_j / G_j, projection $\pi_{i,j}$ is used to identify the output states of intersection components of S_i / G_i, when $\alpha_{i,j}$ occurs.

2.2 Design

Formally, the starting state of mode *n* is given in the form (*q*, *x*), where *q* is the starting process state that will be given by proposition 1, *x* is the starting specification state that will be given by proposition 2. In other words, proposition 1 allows to build the extended model $G_{i,ext}$ for process model G_i. Namely the extended process model for each operating mode $i \in I$ is given by automaton model $G_{i,ext} = (Q_{i,ext}, \Sigma_{i,ext}, \delta_{i,ext}, q_{i,0,ext}, Q_{i,m,ext})$ in which:

- $Q_{i,ext} = Q_i \cup \{q_{i,in}\}$ ($q_{i,in}$ represents the inactive state)
- $\Sigma_{i,ext} = \Sigma_i \cup \Sigma^{'}$
- $q_{i,0,ext} = \begin{cases} q_{i,0} & if\ i = 1 \\ q_{i,in} & if\ i \neq 1 \end{cases}$
- $Q_{i,m,ext} = Q_{m,ext}$
- The extended transition function $\delta_{i,ext}$ is defined as follows:
  - $\forall\ q \in Q_i$ and $\forall\ \sigma \in \Sigma_i$, if $\delta_i(q, \sigma)$ exists then $\delta_{i,ext}(q, \sigma) := \delta_i(q, \sigma)$
  - $\forall\ q \in Q_i$ from which the switching event $\alpha_{i,j}$ can be occurred, then $\delta_{i,ext}(q, \alpha_{i,j}) := q_{i,in}$
  - $\delta_{i,ext}(q_{i,in}, \alpha_{j,i})$ (the set of starting states of model *i*) will be defined according to proposition 1.

Similarly the set of starting state of specification model is determined by proposition 2. Namely for each specification model $E_i = (X_i, \Sigma_i, \xi_i, x_{i,0}, X_{i,m})$ we defined the extended specification model $E_{i,ext} = (X_{i,ext}, \Sigma_{i,ext}, \xi_{i,ext}, x_{i,0,ext}, X_{i,m,ext})$, with:

- $X_{i,ext} = X_i \cup \{x_{i,in}\}$ ($x_{i,in}$ represents the inactive state)
- $\Sigma_{i,ext} = \Sigma_i \cup \Sigma^{'}$
- $x_{i,0,ext} = \begin{cases} x_{i,0} & if\ i = 1 \\ x_{i,in} & if\ i \neq 1 \end{cases}$
- $X_{i,m,ext} = X_{m,ext}$
- The extended transition function $\xi_{i,ext}$ is defined as follows:
  - $\forall\ x \in X_i$ and $\forall\ \sigma \in \Sigma_i$, if $\xi_i(x, \sigma)$ exists then $\xi_{i,ext}(x, \sigma) := \xi_i(x, \sigma)$
  - $\forall\ x \in X_i$ from which the switching event $\alpha_{i,j}$ can be occurred, then $\xi_{i,ext}(x, \alpha_{i,j}) := q_{i,in}$
  - $\xi_{i,ext}(q_{i,in}, \alpha_{j,i})$ (the set of starting states of model *i*) will be defined according to proposition 2.

For more details, the reader could refer to [7].

Proposition 1

Let models $G_1, G_2, \ldots, G_n$ characterize the dynamic process in each operating mode.

1. *Determine a partial function C, defining possible $i-to-j$ switchings in C, if and only if there is a switching from G_i to G_j.*

2. *I = {1}. I represents the set of mode subscripts from which switching events will be considered events, starting from the initial mode.*
3. *While I ≠ {} do:*
   (a) *L = {}. L is a temporary set allowing determination of mode subscripts from which switchings with the following step will be considered.*
   (b) *For each i ∈ I: let L_i be the set of modes such that, for all j in L_i, the i – to – j switching in C.*
      i. *For each G_i such that j∈ L_i:*
         A. *Determine the set of starting states by applying: $\delta_{j,ext}(q_{j,in},\alpha_{i,j}) = \delta_j(q_{j,0},\pi_{i,j}(K_{q,q'}))$* [1] *($\forall s \in K_{q,q'}, \alpha_{i,j} \in follow(s)$* [2]*). This needs to be performed for all $K_{q,q'}$ languages. There are several possible q and q' states.*
         B. *$C = C - \{ i \to j \}$, $i \to j$ represent switching from mode i to mode j.*
      ii. *$L = (L \cup L_i) \cap dom(C)$* [3]
   (c) *do I = L* ♦

The above proposition adopts formally the state from which the model G_i ($i \in \{1,2,...,n\}$) will be activated (the starting state). The following proposition establishes the switching mechanism between specification models by searching the states from which these models must be activated, whilst ensuring adequacy between current process dynamics and control decisions.

Adopting the following notations:

- $\Sigma(q)$ represents the set of generated process events from state q,
- $\Sigma_a(x)$ represents the set of enabled events from specification state x,
- $Re(x,S)$ are the specification states reachable from state x,
- $Re(q,G)$ are the process states accessible from state q.

Proposition 2

Let $q_1, q_2, \ldots, q_n$ be the starting process G_i states.

1. *Determine for each starting state q_i, the desired language K_{q_i} elaborated from this state.*

   Do H = X. Initially H is the set of specification E_i states.
2. *For each q_i do:*
   (a) *Calculate $\Sigma(q_i) \cap K_{q_i}$. This represents the set of process events generated from state q_i and belonging to desired language K_{q_i}.*
   (b) *For each specification state x ∈ H do:*
      i. *Calculate $\Sigma_a(x)$.*

---

1 $K_{q,q'}$ is the language containing all the sequences with starting state q of model S_i / G_i as origin state and a final state like the starting state q' of this model.

2 Denote by *follow*(*s*) the set of events which follow the sequence of events *s*.

3 *dom(C)* represent the field of function *C i.e.* the set of the subscripts *i* such that $i \to j$ belongs to *C*.

ii. *Calculate* $\Sigma_a(x) \cap \Sigma(q_i)$ *. This is the set of process events generated from state* q_i *and enabled from specification state x.*

iii. *If* $\Sigma(q_i) \cap K_{q_i} \neq \Sigma_a(x) \cap \Sigma(q_i)$ *then* $H = H - \{x\}$*. H – {x} is the set H derived of all states x, which do not check the condition.*

iv. *While card(H)* [4] $\neq 1$ *do:*

A. *Calculate Re(x,S).*

B. *Calculate Re(*q_i*,*G_i*).*

C. *If for all x' ∈ Re(x,S) and for all q' ∈ Re(*q_i*,*G_i*), there is an events sequence that checks* $\delta_i(q_i, s) = q'$ *and* $\xi_i(x, s) = x'$*, such that* $s\Sigma(q_i) \cap K_{q_i} \neq s(\Sigma_a(x) \cap \Sigma(q_i))$ *then* $H = H - \{x\}$*.*

v. *State x checking that card(H) = 1 is consequently the unique compatible starting state* q_i *of specification model.* ♦

The previously established proposition makes it possible to complete building the extended controlled process for each operating mode *i*. In the following, we define in formal terms wide models ($S_{i,ext}$ / $G_{i,ext}$) for each operating mode *i*: the extended controlled process model for each operating mode $i \in I$ is given by automaton model $S_{i,ext}$ / $G_{i,ext}$ defined formally by: $S_{i,ext}$ / $G_{i,ext}$ = $S_{i,ext} \times G_{i,ext}$ = {$X_{i,ext} \times Q_{i,ext}$, $\Sigma_{i,ext}$, $\xi_{i,ext} \times \delta_{i,ext}$, ($x_{i,0,ext}$, $q_{i,0,ext}$), $X_{i,m,ext} \times Q_{i,m,ext}$} in which:

- $X_{i,ext} \times Q_{i,ext} = X_i \times Q_i \cup (x_{i,in}, q_{i,in})$,
- $\Sigma_{i,ext} = \Sigma_i \cup \Sigma_i^{'}$ where $\Sigma_i^{'}$ is the set of events allowing to leaving or returning to mode i,
- $(x_{i,0,ext}, q_{i,0,ext}) = \begin{cases} (x_{i,0}, q_{i,0}) & \text{if } i = 1 \\ (x_{i,in}, q_{i,in}) & \text{if } i \neq 1 \end{cases}$
- $X_{i,m,ext} \times Q_{i,m,ext} = X_{i,m} \times Q_{i,m}$,
- extended transition function is given as follows:
  1. $\forall (x, q) \in X_i \times Q_i$ and $\forall \sigma \in \Sigma_i$, if $\xi_i \times \delta_i((x, q), \sigma)$ exists (*i.e.* $\xi_i(x, \sigma)$ exists and $\delta_i(q, \sigma)$ exists), then $\xi_{i,ext} \times \delta_{i,ext}((x, q), \sigma) = \xi_i \times \delta_i((x, q), \sigma)$
  2. all other transitions will be determined by using the proposition 1 and proposition 2.

2.3 Forbidden compatible states

In this section, we study the problem of switching from states in which compatible states in the selected mode are ghost (these states are called ghost, because their existence would potentially violate the defined selected mode specification). For the sake of brevity, a controlled process state will be denoted by *y*. To ensure better understanding and uphold intuitively the concept, only 2 modes will be considered in the following section. As denoted in the previous section, each operating mode is represented by a process model assigned with a specification model. We recall that our contribution above is an algorithm which generates a set of compatible connection states between modes. Specifically, we have shown that if we leave controlled process S_i / G_i from a

4 *card*(*H*) represents the number of elements in *H*.

state y, we must thereby activate the controlled process S_j / G_j from a state y', such that y' is compatible with y. However, the problem is what will happen when state y' is ghost in the controlled process S_j / G_j ?
To grasp our proposition, let us consider the following example.

❑ *Example*

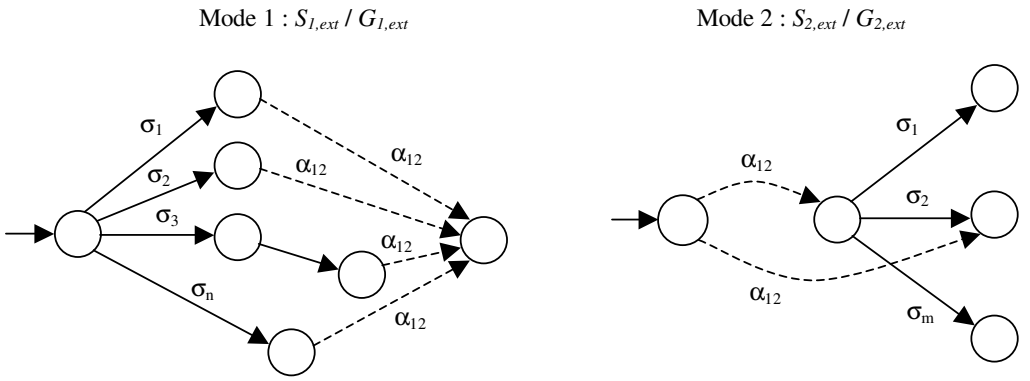

Figure 2: Example of application

Assuming that initially only mode 1 is activated, so from $y_{1,0}$, occurrence of event σ_1 leads S_1 / G_1 to state y_1, in which switching event $\alpha_{1,2}$ is possible. Switching event $\alpha_{1,2}$ can occur in several states of model S_1 / G_1: y_1, y_2, y_4 and y_n. When this event occurs, model $S_{1,ext} / G_{1,ext}$ enters state $y_{1,in}$ (proposition 1 and 2). On the other hand, the set of compatible connection states of y_1 and y_2 in mode 2 are assumed to be $y_{2,0}$ and $y_2^{'}$ respectively. However, when switching event occur from state y_4 and y_n, their compatible connection states in mode 2 do not exist, so y_4 and y_n are forbidden. In this example, we have illustrated only the problem of switching from states in mode 1, in which their compatible connection states in mode 2 are ghost. We can encounter the same problem on switching from mode 2 to mode 1. ❑

Based on Kumar's algorithm [9], we suggest a methodology for ensuring switching between enabled compatible connection states. For each operating mode i, the strategy adopted can be informally described in proposition 3. However, we must firstly give the formal definition of forbidden and pre-forbidden states.

Definition 2

A state y is called a:

1. *Forbidden state if and only if:*
   - *the switching event can occur from y,*
   - *the compatible state of y doesn't exist in the reachable selected mode.*
2. *Pre-forbidden state if and only if:*
   - *the switching event can't occur from y,*
   - *there is a sequence of uncontrollable events , whose occurrence leads to a forbidden state.* ♦

Proposition 3

Step 1: *calculate controlled process S_i / G_i ($L(S_i / G_i)$ is assumed controllable with respect to G_i)*

Step 2: *identify all forbidden states BS(mode i)*

Step 3: *identify all pre-forbidden states PBS(mode i)*

Step 4: *delete from S_i / G_i all states in BS(mode i) and PBS(mode i) (also all transitions associated with these states)*

Step 5: *delete all states y of S_i / G_i from which there are no paths to y from the initial state of S_i / G_i .* ♦

A controllable event leading to either a forbidden state or a pre-forbidden state can be directly disabled. On the other hand, in the case of an uncontrollable event leading to forbidden state, we therefore disable the controllable event leading to the state from which the sequence of uncontrollable events can be occur. The language obtained in this way is controllable. There is therefore a supervisor achieving this language. The problem of calculating this supervisor has been omitted from this paper.

Remark 1

It should be remembered that this approach makes it possible to switch only between existing compatible states enabled in two operating modes. It does however restrict, in terms of permissivity, the controlled process behavior in these two operating modes. ♦

3 Numerous operating mode switching

Proposed definition of multiple forbidden or pre-forbidden starting states in operating mode management allows implementation of more significant switching laws. Industry indeed requires component accommodation capacity in certain tangible applications prior to mode transition.

The following example effectively illustrates this by considering a generation unit supplying power to a user net after the failure of the nominal energy provider. This switching action is clearly indirect and requires component accommodation capacity. Switching from one operating mode to another requires not only several transient modes, but also definition of acceptable trajectories, depending on the failure state detected early in the control design phase.

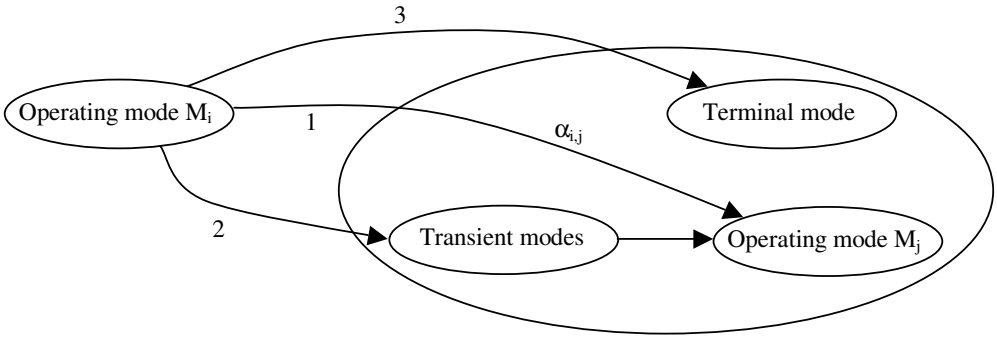

Figure 3: Different possible trajectories linking operating mode M_i to operating mode M_j

Based on component or control flexibility, stabilisation capacity is introduced to propose a generic method of undertaking this form of mode management. From the control standpoint, a forbidden starting state defines a repulsive trajectory and so a pre-forbidden starting state is therefore associated with attractive trajectories. The challenge here is to provide control, which maintains starting states that can be stabilised in relation to attractive

action or, in other words, exclude such starting states in relation to repulsive actions. This implies existence of a set of controllable trajectories permitting the new prescribed operating mode to be reached in a finite iteration process. Both forbidden and pre-forbidden states are thereby generalised and switching laws governing Discrete Event System operating modes are formally expressed though stabilisation control.

Figure 3 illustrates three different trajectories under switching. The state, in which the switching event has occurred, effectively governs these trajectories:

- **case 1:** direct switching from operating mode M_i to operating mode M_j is possible and does not require component adjustment;
- **case 2:** no direct switching from operating mode M_i to operating mode M_j is possible and component adjustments are required; transient modes must be prescribed to access operating mode M_j;
- **case 3:** switching from operating mode M_i to operating mode M_j is impossible; a terminal operating mode is specified; in practice, this case represents the safe mode. Different attractiveness or repulsiveness typologies can be defined. These are associated with the activated component operating rate based on the following operating mode definition..

3.1 Organic definition for operating mode

Operating mode M_x is described as a stable configuration, in which n components R_i are activated to fulfil an attempted task T_x through interaction $L_{i,j}$ with a component R_j. The activation level of each component j is identified by its charge $r_{j,y}$.

Formally, the organic model for the mode M_x is $M_x = (R_i, r_{j,y}, L_{i,j}, T_x)$ for $i = 1, \ldots, n$, $j = 1, \ldots, n$, $y \in \{Min, Med, Max\}$ and $x = 1, \ldots, m$. Considered components are physical entities acting on other physical entities. Operating rate $r_{j,y}$ will be considered for a component j at three levels (minimum, medium, maximum or *Min*, *Med*, *Max* respectively) to simplify the proposed method. Operating rate $r_{j,y}$ can be constant or variable for a given mode M_i, depending on the involved task T_x.

Components R_i are connected to each other by a functional interaction $L_{i,j}$. According to the literature [13], these interactions appear explicitly as Input ($In - R_i$), Output ($Out - R_i$) and Control ($Cont - R_i$) interfaces. Input or output external signals will be called Ix_y or Ox_y. Figure 4 illustrates a stable operating mode M_1, in which three components are activated at operating rates $r_{1,Med}$, $r_{2,Med}$ and r_3 (varying between $r_{3,Med}$ and $r_{3,Max}$) respectively. Interaction links are (Ix_{11} / $In - R_1$) and ($Out - R_1$ / $Cont - R_2$) for R_1, (Ix_{12} / $In - R_2$), ($Out - R_1$ / $Cont - R_2$) and ($Out - R_2$ / $In - R_3$) for R_2 and ($Out - R_2$ / $In - R_3$) and ($Out - R_3$ / Ox_1) for R_3.

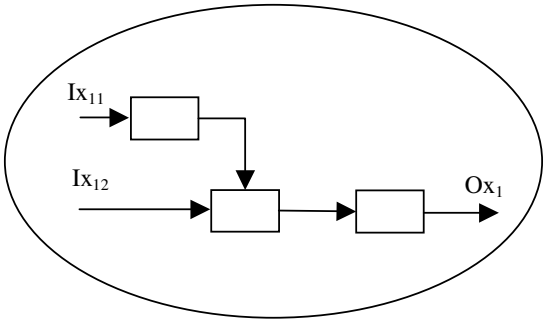

Figure 4: Organic configuration for M_1

Mode switching is allowed if and only if there is a sufficient degree of freedom in component control terms. Mode switching will lead to interaction links reconfiguration and a new set of activated components, rather than deactivation of certain components in the previous configuration. We now consider an attempted, new stable operating mode M_2, in which R_4 and R_5 replace the defective R_3. We assume that R_1 and R_2 are common components implemented in modes M_1 and M_2 at the same activation level.

Figure 5 presents the new organic configuration with R_1 activated to identical $r_{1,Med}$, R_2 activated to $r_{2,Med}$, whilst the new components R_4 and R_5 are activated at activation rates $r_{4,Med}$ and $r_{5,Med}$ respectively. Interface links in M_2 have become $(Ix_{11} / In - R_1)$ and $(Out - R_1 / Cont - R_2)$ for R_1, $(Ix_{12} / In - R_2)$, $(Out - R_1 / Cont - R_2)$ and $(Out - R_2 / In - R_4)$ for R_2, $(Out - R_2 / In - R_4)$ and $(Out - R_4 / In - R_5)$ for R_4 and $(Out - R_4 / In - R_5)$ and $(Out - R_5 / Ox_2)$ for R_5.

Within this reference frame, switching $M_1 - M_2$ prompts component, interface and activation rate changes. We can readily assume that such switching are indirect and this assumption can be accepted even more readily, when the new implemented component cannot alone provide the activation rate of the defective component in M_1. The following switching trajectories can be defined based on the three previous switching cases, depending on failure of R_3 at activation rate $r_{3,Med}$ or $r_{3,Max}$.

We consider it is impossible to commute to operating mode M_2, if switching is necessary when R_3 operates at activation rate $r_{3,Max}$ in operating mode M_1. The process will be performed in a terminal mode M_{term1}. The starting states resulting in terminal mode M_{term1} are defined as repulsive states.

On the other hand, if switching is necessary when R_3 operates at activation rate $r_{3,Med}$ in operating mode M_1, this can be performed by successively activating R_4 and R_5 at identical activation rates $r_{4,Med}$ and $r_{5,Med}$. Initial implementation of R_4 establishes a transient mode M_{trans1}. Starting states in transient operating mode M_{trans1} will be defined as attractive states.

Figure 6 illustrates the different switching trajectories when $M_1 - M_2$ switching is needed. It assumes that component pre-configuration be performed, when direct switching is forbidden or is technically impossible.

A mode is defined as a transient mode when a controllable or uncontrollable trajectory leads to the attempted

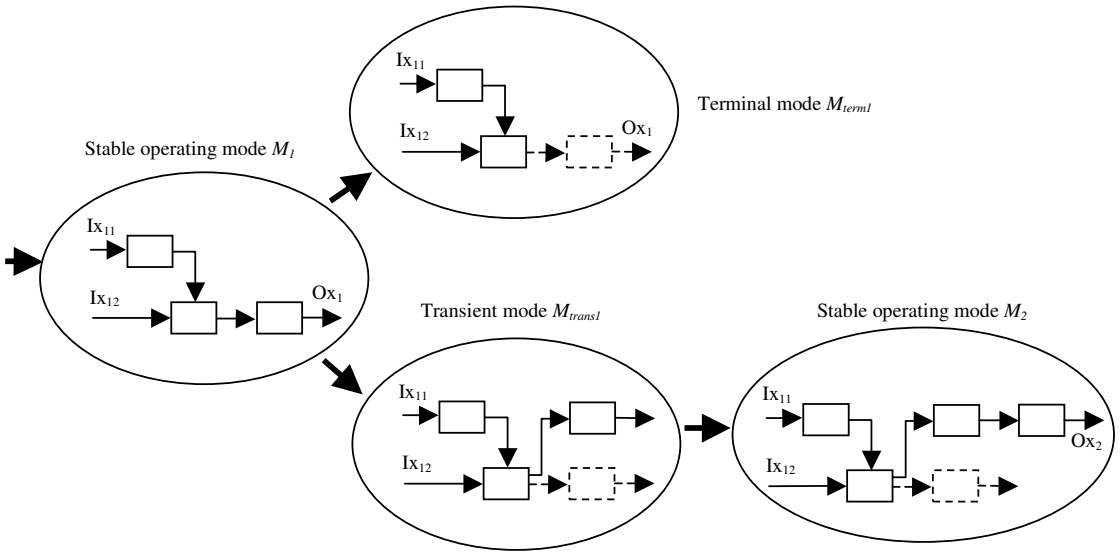

Figure 5: Switching with a transient mode and a terminal mode

next stable operating mode in a finite number of events. In this way, the set of the permitted trajectories results in a stabilisation switching control law. A mode is defined as a terminal mode when a controllable or not uncontrollable trajectory cannot lead to the attempted, next stable operating mode. A terminal mode is a dead mode, requiring in-depth initialisation of all components. An operating mode will be defined as a stable mode until a switching event occurs.

The behavioral model of the operating mode is defined as $M_i = \{Q_i, \Sigma_i, \delta_i, q_{0,i}, Q_{m,i}\}$, where Q_i is the state space, Σ_i is the alphabet, δ_i is the transition partial function, $q_{0,i}$ is the initial state and $Q_{m,i}$ is the subset of marked states of the operating mode M_i.

Lets Q_{transi} the states space of transient mode M_{transi}, Σ_{transi} the alphabet of transient mode, δ_{transi} the transition partial function of transient mode and Q_i the state space of an operating mode M_i. The length $|s|$ of a string $s \in \Sigma^*$ is defined according to $|s| = k$ if $s = \sigma_1 \sigma_2 \ldots \sigma_k$. The formal definition of a transient mode M_{transi} is as follows: $\forall q \in Q_{transi}$, $\exists\, s = \sigma_1 \sigma_2 \ldots \sigma_n$ with $\sigma_l \in \Sigma_{transi}$ $(l = \{1, \ldots, n\}$ such that $\delta_{transi}(q, s) \in Q_i$ or $\delta_{transi}(q, s) \in Q_{transj}$ $(i \neq j)$.

Let Q_{term} the states space of terminal mode M_{term}, Σ_{term} the alphabet, δ_{term} the transition partial function and Q_{op} the states space of this operating mode. The formal definition of formal terminal mode M_{term} is as follows: $\forall q \in Q_{term}$, it doesn't exist σ such that $\delta_{term}(q, s) \notin Q_{term}$.

3.2 Switching laws with attractive/repulsive starting states

A switching law implies that a set of successive operating modes could be activated in relation to high-level specifications. It also defines the capacity of the process to evolve from one component configuration to another. These assumptions result in design of explicit switching control. In control problem terms, we simply characterise jump trajectories at this stage; stabilisation represents one control feature. A stabilisation switching control law can indeed be established for the operating mode which can be reached. This law is based on a set of trajectories, when jumping from one operating mode to another. Trajectory definition is of course necessary and we therefore distinguish a trajectory, which includes starting states belonging to a transient mode, from a trajectory, which includes starting states belonging to a terminal mode. Starting states will be defined as attractive states in the former mode and as repulsive states in the latter mode.

Application: uninterrupted electrical power distribution

We consider two independent electrical generators (main R_1, secondary/back-up R_2) supplying power to a set of users through a connector C. The main generator fails (we assume the secondary/back-up generator remains unaffected by this failure) and the aim is to activate (via a component *start*) a back-up generator to maintain an identical level of supply to users. The back-up generator comprises two separate, independently powered separate units G_1 and G_2.

The problem considered is to ensure an uninterrupted service at the same power generating rate. We naturally assume that direct switching from nominal mode M_1 to rescue one M_2 is impossible.

The rescue procedure is described as follows: if the nominal mode generating rate is acceptable for power generator activation (acceptable for newly activated components), G_1 is initially activated and is followed by G_2, if the G_1 generating rate is high enough. Switching to concurrent power supply offers at least one unique solution

(M_{term}) for other situations, in which the nominal mode generating rate does not permit the generator activation or in which G_1 activation does not reach the required generating rate.

Specified operation considers that the system commutes by activating G_1 followed by G_2 (first transient mode M_{1rans1}) after failure of component R_1 in nominal mode M_1 (the detection is based on its generating rate r_{R1}). Thereafter, if G_1 supplies attempted power r_{G1}, G_2 is newly activated (second transient mode M_{1rans2}). The latter switching accesses desired operating mode M_2 as long as both G_1 and G_2 operate properly, as reflected by generating rates r_{G1} and r_{G2}.

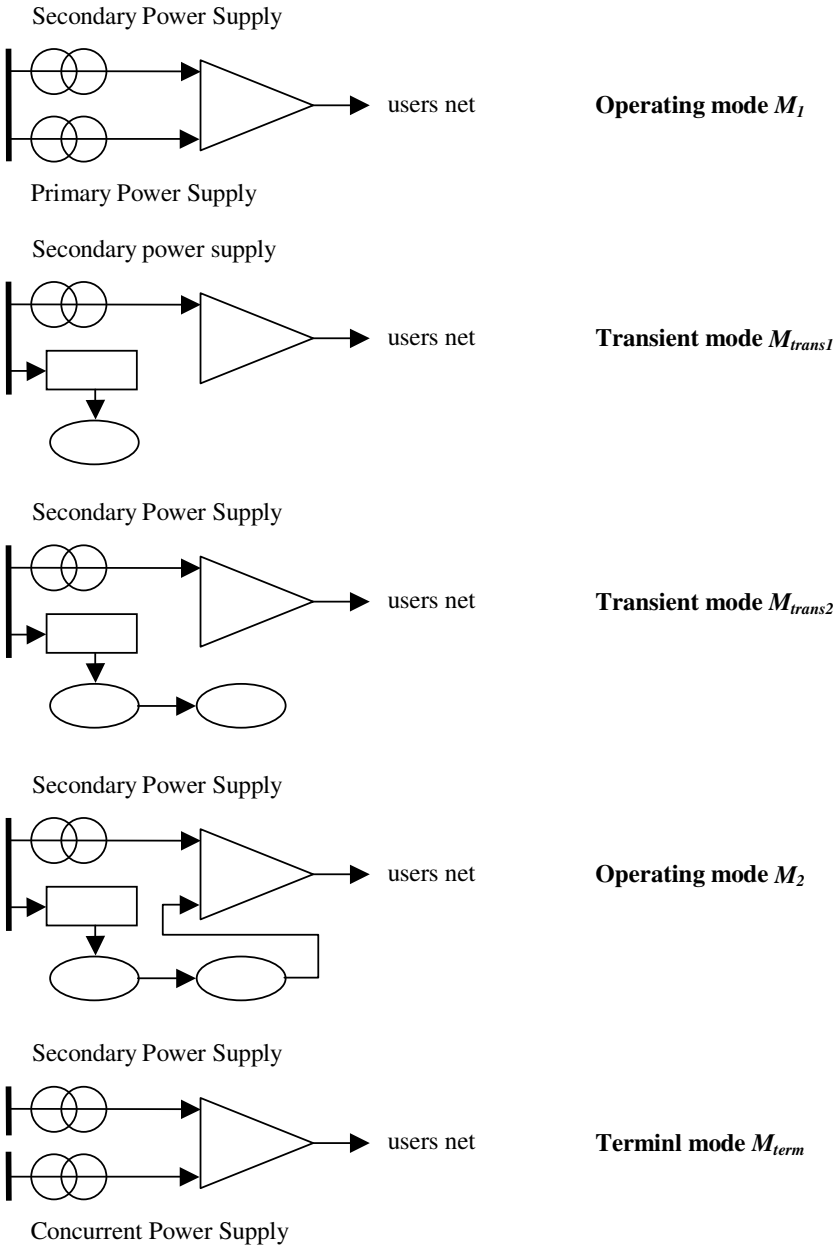

Figure 6: Organic representation of the 5 modes

Common components are present in each transient, operating or terminal mode. In this example, R_1 and C are the main common components and, under these circumstances, previously described computation of each starting state is necessary.

The figure 6 consolidates the five different organic configurations.

3.3 Switching trajectories

Based on the allowed switching (figure 7), different trajectories could be defined as $M_1 - M_{term}$, $M_1 - M_{trans1}$, $M_{trans1} - M_{term}$, $M_{trans1} - M_{trans2}$, $M_{trans2} - M_2$. An attractive state belongs to the switching trajectory set [$M_1 - M_{trans1} - M_{trans2} - M_2$] and repulsive states belong to both [$M_1 - M_{term}$] and [$M_1 - M_{trans1} - M_{term}$] switching trajectory sets.

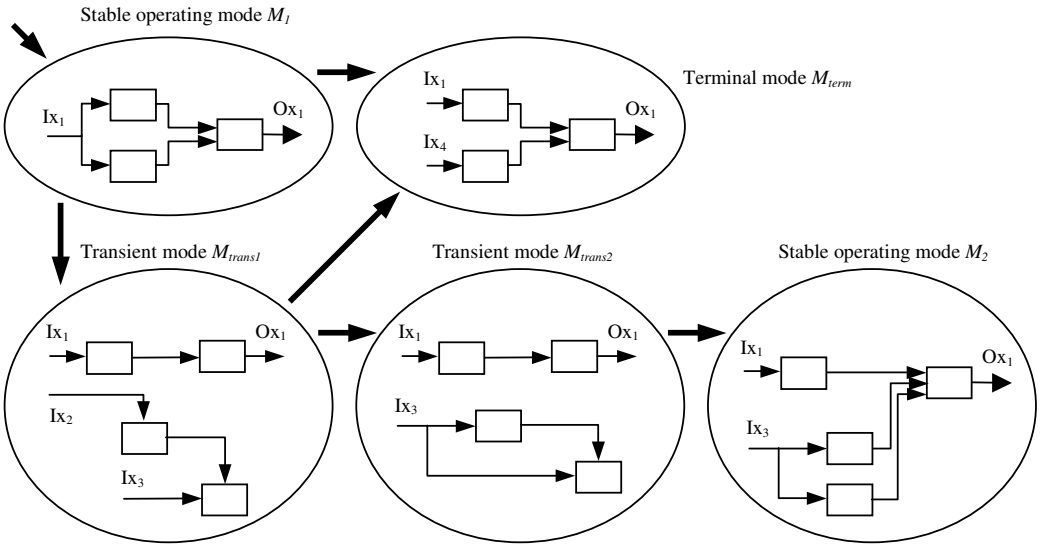

Figure 7: Trajectory operating mode M_1 – operating mode M_2

The organic structure relating for trajectory $M_1 - M_2$ is illustrated in tables 1 and 2.

Modes	Component	Comment	Power rate	Links
M_1	*PPS*	primary power supply	Med – Max	Ix_1 / *In – PPS* *Out – PPS* / *In – C*
	SPS	secondary power supply	Med – Max	Ix_1 / *In – SPS* *Out – SPS* / *In – C*
	C	users connector	non defined	*Out – PPS* / *In – C* *Out – SPS* / *In – C* *Out – C* / Ox_1
M_{trans1}	*SPS*	secondary power supply	Med – Max	Ix_1 / *In – SPS* *Out – SPS* / *In – C*
	C	users connector	non defined	*Out – SPS* / *In – C* *Out – C* / Ox_1
	St	Starter	non defined	Ix_2 / *Cont – St* *Out – St* / *Cont* – G_1
	G_1	first generator	Med	Ix_3 / *In* – G_1 *Out – St* / *Cont* – G_1

Table 1: Switching from operating mode M_1 to M_{trans1}

Modes	Component	Comment	Power rate	Links
M_{trans2}	*SPS*	secondary power supply	Med – Max	Ix_1 / *In – SPS* *Out – SPS / In – C*
	C	users connector	non defined	*Out – SPS / In – C* *Out – C / Ox*$_1$
	G_1	first generator	Med	Ix_3 / *In* – G_1 *Out* – G_1 / *Cont* – G_2
	G_2	second generator	Med	Ix_3 / *In* – G_2 *Out* – G_1 / *Cont* – G_2
M_2	*SPS*	secondary power supply	Med – Max	Ix_1 / *In – SPS* *Out – SPS / In – C*
	C	users connector	non defined	*Out –SPS / In – C* *Out* – G_1 / *In – C* *Out* – G_2 / *In – C* *Out – C / Ox*$_1$
	G_1	first generator	Med	Ix_3 / *In* – G_1 *Out* – G_1 / *In – C*
	G_2	second generator	Med	Ix_3 / *In* – G_2 *Out* – G_2 / *In – C*

Table 2: Switching from operating mode M_{trans2} to M_2

External inputs can differ: in this case, Ix_2 represents generator starter activation prompted by failure of component *PSP*. Certain generating rates don't need to be described, such as that of connector *C* and starter *St*. Common components are explicitly described by functional dependencies.

Attractiveness and repulsiveness must now be expressed in control terms. From the control standpoint, the main aim is to define the resulting consistent trajectories, including starting states, based on the previously described multimodeling approach.

For a trajectory including attractive states, computation of starting states in each transient mode is based on the stability characteristic. Starting states in each transient mode are computed such that the subsequent stable operating mode is accessible in a finite number of steps.

For a trajectory including repulsive states, computation of starting states in each transient mode is based on the livelock characteristic. Starting states in each transient mode are computed such that the subsequent accessible operating mode is a livelock mode.

Formally the attractive states set is defined as follows: $Q_A = \{q \in Q_{transj} \mid \exists\ \alpha_{i,transj}\ ,\ \delta_{transj,ext}\ (q_{transj,in},\ \alpha_{i,transj} = q\}$, such that $\alpha_{i,transj}$ represents the switching event form operating mode M_i to transient mode M_{transj}, $\delta_{transj,ext}$ the extended transition function of transient mode (see 2), $q_{transj,in}$ the inactive state of transient mode and Q_{transj} the state set of transient mode.

The repulsive set is defined formally as follows: $Q_R = \{q \in Q_{term} \mid \exists\ \alpha_{i,term}\ ,\ \delta_{term,ext}\ (q_{term,in},\ \alpha_{i,term} = q\},\ \alpha_{i,term}$ represents the switching event from operating mode M_i to terminal mode M_{term}, $\delta_{term,ext}$ the extended transition function of terminal mode, $q_{term,in}$ the inactive state of terminal mode and Q_{term} the transition function of terminal mode.

4 Conclusion

This paper proposes a Supervisory Control Theory-based approach. We have presented a framework for managing switching of systems, whose dynamics change as they evolve in several operating modes. Our primary contribution is the introduction of a multi-model approach involving representation of a complex system by

several simple models. Each model is a partial description of the system in a given operating mode. Initially, only one model is activated and the nominal operating mode is generally assumed. All other modes are effectively deactivated. Common components are possible in each considered mode and the concept of tracking is introduced. We have therefore extended each considered controlled process model and defined a switching mechanism, which makes it possible to track explicitly the behavior of each process model. This switching mechanism is characterised by information channels. In other words, we have shown that switching between modes is only between compatible states. We have shown also that there is a subset Q of states in mode i (resp. in mode j) from which the switching event can occur and that their compatible connection states in mode j (resp. in mode i) are ghost. We have therefore proposed an algorithm permitting avoidance of both this subset of so-called forbidden states and of the set of so-called pre-forbidden states of mode i (resp. of mode j), from which the occurrence of the uncontrollable event sequence leads to a forbidden state of Q (resp. of Q').

Attractive and repulsive states have been defined by considering transient and terminal operating modes respectively. These definitions introduce stability and livelock characteristics in switching laws. Operating mode management can thus be discussed in process control terms. Specifying switching could provide the desired requirements and characteristics could validate those requirements.

Current research is attempting to optimize these switching trajectories based on the consumption and cost of the newly implemented component demanding supply.

References

1. Adepa, "Guide d'Etude des Modes de Marches et d'Arrêts (GEMMA)", 1981.
2. Asarin, E., Bournez, O., Dang, T., Maler, O. and Pnueli, A. "Effective Synthesis of Switching Controllers for Linear System", Proceedings of IEEE, vol. 88, pp. 1011-1025, 2000.
3. Chafik, S. and Niel, E. "Hierarchical-decentralized solution of supervisory control", 3rd International Symposium on Mathematical Modeling, 3rd MATHMOD, vol. 2, pp. 787-790, Wien, Austria, 2000.
4. Charbonnaud, P., Rotella, F. and Médar, S. "Process Operating mode Monitoring Process: Switching Online the Right Controller", IEEE Transactions on Control Systems Technology, vol. 31, pp. 77-86, 2002.
5. Hamani, N., Dangoumau, N. and Craye, E. "A formal approach for reactive mode handling", IEEE international conference on Systems, Man and Cybernetics, SMC04, pp. 4306-4311, The Hague, Netherlands, 10-13 October 2004.
6. Hashtrudi Zad, S., Kwong, R. H. and Wonham, W. M. "Fault Diagnosis in Discrete-Event Systems: Incorporating Timing Information", IEEE Transactions on Automatic Control, vol. 50, n°7, pp. 1010-1015, 2005.
7. Kamach, O., Piétrac, L. and Niel, E. "Generalisation of Discrete Event System multi-modeling", 11th IFAC Symposium of Information Control Problems in Manufacturing, INCOM'04, 6 p, Salvador-Bahia, Brazil, 5-7 April 2004.
8. Kamach, O., Piétrac, L. and Niel, E. "Supervisory Uniqueness for Operating Mode Systems", 16th IFAC World Congress, 6 p, Prague, Czech Republic, 4-8 July 2005.
9. Kumar, R. "Supervisory synthesis techniques for discrete event dynamical systems", PhD thesis, University of Texas, USA, 1991.
10. Lin, F. and Wonham, W. M. "Decentralized control and coordination of discrete-event systems with partial observation", IEEE transactions on automatic control, vol. 44, pp. 1330-1337, 1990.

11. Nourelfath, M. and Niel, E. "Modular supervisory control of an experimental automated manufacturing system", Control Engineering Practice, vol. 12, n°2, pp. 205-216, 2004.
12. Ramadage, P. J. G. and Wonham, W. M. "Control of discrete-event systems", IEEE transaction on automatic control, vol. 77, pp.81-98, 1989.
13. Ross, D. T., Kenneth, E. and Schoman, J. "Structured Analysis for Requirements Definition", IEEE transactions on Software Engineering, vol. SE-3, n°1, pp. 86-95, 1977.
14. Rudie, K. and Wonham, W. M. "Think globally, act locally: decentralized supervisory control", IEEE transactions on automatic control, vol. 37, pp. 1692-1708, 1992.
15. Wong, K., Thistle, J. G., Malhame, R. and Hoang, H. "Supervisory Control of distributed Systems: conflict resolution", Discrete Event Dynamic Systems: theory and applications, vol. 10, pp. 131-186, 2000.
16. Zefran, M. and Burdick, J. "Design of switching controllers for systems with changing dynamics", 37th Conference on Decision and Control ,CDC, pp. 2113-2118, Phoenix, Arizona, USA, December 1998.