Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A framework for FSM based multi-model approach to interconnected components' network

Babiga Birregah
University of Technology of Troyes
Charles Delaunay Institute
Laboratory of Industrial Systems Optimisation
Troyes - Framce

Kondo Hloindo Adjallah, Member, IEEE
Ecole Nationale d'Ingnieurs de Metz
Laboratory of Industrial and Production Engineering
Metz - France
adjallah@enim.fr

***Abstract*—This paper presents a framework of multi-model based approach for interconnected components network by finite state machine model. The main concept on which we base our approach is the fact that, in a context of interconnected components, due to the influence of some components named here key-components the system can switch from one mode to another. The influence of these components can be measured by the level of involvement in the switching events. The proposed approach can be applied to embedded systems in which we propose to consider the controller as a part of the system.**

***Index Terms*—Interconnected components system, Level of involvement, Key-component, Multi-model, Finite states machine.**

I. Introduction

One of the challenges of this last decade, which arises in discrete-event systems (DES) studies, is the possibility that a system can change its operating mode while performing its mission. In some systems the set of events is splited into two subsets [7]. The first one is the set of events triggered by the environment of the system. The second one is the set of events conditioned by the state of the system's components. This is particulary true for dynamic systems. In both cases the main concern of researchers is to identify and control the source of mode switching.

This present work proposes an attempt to address this problem under the assumption that all events in the finite state machines (FSMs) modeling the components in an interconnected components' network (INC) are trigged by at least one of these components. As in most models proposed for modes' switching [1], we propose a framework to characterize a mode. But our model is motivated by the crucial problem of commutation events, not much studied in the literature. To take into account mutual influence events we introduce a set of special components, called key-component, which play an important role in the switching events.

II. Behavior-Based FSM Model for ICN Systems

A. FSM Model for a component in an ICN System

Let us consider a distributed system of n interconnected components denoted C_k. For $k = 1, 2, ..., n$, component C_k will be modeled by a Finite State Machine (FSM) $\mathcal{A}_k = (\Sigma_k, S_k, I_k, \Delta_k, F_k)$. In general, in finite state machine modeling a given component C_k, the set of input alphabet Σ_k consist in local events taking place in this component. This model is appropriate for components working independently from each other. In a context of interconnected components systems, the state or the performance of a target component may depend on the state of the others. This is the case in grid network or ICNs in which the components, not only share the same resources, but also interact. A change of state of a given component can be influenced by the state of the others or by informations from the others.

For example, an electric power network is composed of an electric power source station, a power converter with regulator and an electric load. The three subsystems function in a close loop, and have mutual influences on each other. Any variation in a subsystem parameter of its involved component may lead to change of the whole systems dynamics to a specific mode. In addition, this change of dynamics can also trigger, through successive reactions, the change of a parameter of another subsystem. This type of interactions and dynamics was studied by Monovskaya et al [8] to monitor degradation and predict failure of pulse energy conversion systems based on bifurcation analysis dynamics fractal regularities interpretation. The major issue in their approach was to map the system's dynamics in the space of state vs parameters. This is a first step in deriving accurate models that easily handle the dynamics of the system's interactions.

In general the whole interconnected components network will be modeled by the FSMs product:

$$\mathcal{A} = \prod_{k=1}^{n} \mathcal{A}_k = \left(\prod_{k=1}^{n} \Sigma_k, \prod_{k=1}^{n} S_k, \prod_{k=1}^{n} I_k, \prod_{k=1}^{n} \Delta_k, \prod_{k=1}^{n} F_k \right).$$

The particularity of this model is that the set $\prod_{k=1}^{n} \Sigma_k$ of events can help to evaluate the involvement of each component in a change of state as a result of an event. Since it is assumed that we can know the component which motivates each event of Σ_k, it is possible to assess the contribution of each component in the event vectors in the set $\prod_{k=1}^{n} \Sigma_k$.

For each component C_k, the set Σ_k will be partitioned as follows:

$$\Sigma_k = \bigcup_{i=1}^{n} \Sigma_k^{(i)},$$

where $\Sigma_k^{(i)} = \left\{\sigma_{k,j}^{i}, \quad j = 1, 2, ...\right\}$ is the set of events on C_k conditioned by the behavior of C_i. For convenience the events $\sigma_{k,j}^{k}$ of the set of inner events $\Sigma_k^{(k)}$ will be denoted $\sigma_{k,j}$. If the behavior of C_i does not affect C_k then $\Sigma_k^{(i)}$ is empty.

Definition 1: Level of involvement (LI)

For a given event $\sigma = \left(\sigma_{1,j_1}^{i_1}, \sigma_{2,j_2}^{i_2}, ..., \sigma_{n,j_n}^{i_n}\right) \in \prod_{k=1}^{n} \Sigma_k$, let us consider the vector $u = (i_1, i_2, ..., i_n)$. The *level of involvement* (LI) of the component C_i in the transition labeled by σ is $Inv(i, \sigma) = \frac{1}{n} \sum_{k=1}^{n} \delta_{u_k}^{i}$.

As can be seen:

$$0 \leq Inv(i, \sigma) \leq 1.$$

$$\text{Obviously, } Inv(i, \sigma) = \begin{cases} 0 & \text{if } C_i \text{ is not involved in } \sigma, \\ 1 & \text{if } C_i \text{ is the only component involved in } \sigma. \end{cases}$$

In distributed systems with large number of quasi-autonomous components, few components will have their level of involvement closed to 1. These components can be assimilated in some cases to the system's controllers or supervisors. But, in a system involving a large number of components with strong dependency, the LI of several components will be high. This will be the case in several fault tolerant embedded systems, reconfigurable systems, ...

Autonomy and dependency are two challenges that large distributed system have to deal with. In dependency case, the LI can help to evaluate the synchronization level of the components. For example, in high synchronous components system, for a given transition σ, all $Inv(i, \sigma)$ will be sensibly equal.

The FSM model presented in this paper allows the study of a system simply just considering its different global behavior and mode's switching events.

By *behavior* we mean in this paper the black box description of a component concerning the relationship between the input information and the outputs [3]. Thus one can say that a first way to characterize a system behavior is to refer to the class of states that are involved in it [2].The principle of the present approach is to track behavior of components involved in an interconnected componets' network by using FSMs model.

B. Behavior-Based Model

1) Basic concepts: Usually the state of a whole system as a *combination* of components local states is coded into an observation. Each class of observations is closely related to a behavior that results in a particular function mode of the system. The state of components involved in a given behavior of the system may change while this behavior remains the same. Two cases may arise:

a) the new state of a component can be a state already involved in the current behavior,

b) or the new state can be different from all states involved in the current behavior.

In both cases the change of state can lead to a change of function mode or behavior. Figure 1 shows the case of two components system which change mode when the state of component modeled by FSM_2 changes. In this example we say that component modeled by FSM_2 is a *key-component* in the transition from mode 1 to mode 2.

Definition 2: Key-component

A component U is a *key-component* in the transition from model r to model s, if there exists a non-empty set of states of U denoted $KeySt(U, r, s)$ such that the system changes its behavior form mode r to mode s when this component reaches one of the states of $KeySt(U, r, s)$.

It is readily seen that for each transition σ outgoing from model r to reach model s, the level of involvement $Inv(U, \sigma)$ of a key-component is more significant than others.

Let us assume that each mode is known by its set of states and that the mode r is the current mode. To verify that U is a key-component one varies the local state of this component in the states represented in the mode r. The resulting states are then compared with the states of the modes s. U will be a key-component if it matches with a state of s and especially if this state is reachable from a state of r. This method requires that the two modes must be known by their exact automaton. But in practice it is difficult to know a mode by all its states.

We call $KeySt(U, r, s)$ the set of *Key-states* of the component U in the transition of the system from r to s.

In the following we will refer to $KeyCp(r, s)$ as the set of key-components involved in the switch from model r to model s. In general, if they have any meaning,

$$KeySt(U, r, s) \neq KeySt(U, s, r), \tag{1}$$

$$KeyCp(r, s) \neq KeyCp(s, r). \tag{2}$$

Equation 1 states that if U is a key-component in the switches $r \rightarrow s$ and $s \rightarrow r$, the state by which the system leaves the mode r can be different from the one by which it returns to this mode. This reflects the fact that a mode of operation can be permanent or temporary. A mode r will be said to be permanent if the problem:

$$(P_1) \begin{cases} \textit{Find } s \\ \textit{such that } KeyCp(r, s) \neq \emptyset \end{cases}$$

does not have a solution. Denoting by Mod the set of all modes of the system, this turns into checking if the set $KeyCp(r) = \bigcup_{s \in Mod} KeyCp(r, s)$ of component potentially able to exit the system from this mode is empty.

In general, a system can not be in two modes simultaneously

unless there are non-distinguishability constraints. But (P_1) can have more than one solution. In this case these solutions do not occur simultaneously or they can not be caused by the same switching event.
It follows from what precedes that:

$$U \in KeyCp\,(r, s) \;\Leftrightarrow\; KeySt\,(U, r, s) \neq \emptyset$$

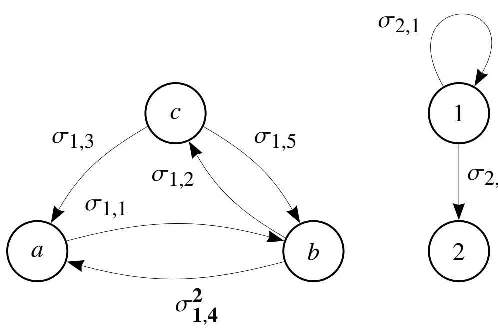
Considering two particular modes r and s, the set Cp of all components in the system is partitioned as

$$Cp = KeyCp(r, s) \cup \overline{KeyCp(r, s)}.$$

More generally,

$$Cp = KeyCp \cup \overline{KeyCp},$$

where $KeyCp = \bigcup_{r,s \in Mod} KeyCp\,(r, s)$ is the set of all key-components of the system.



(a) FSM of C_1 (b) FSM of C_2

	a	b	c
1	*Mode r*	*Mode r*	*Mode r*
2	*Mode s*	*Mode r*	*Mode s′*

(c) Table of modes of the system

Fig. 1. Function Mode changing by a change in component's state

For example let us consider a two components system represented by $FSM_i = (\Sigma_i, S_i, I_i, \Delta_i, F_i)$, $(i = 1, 2)$, in figure 1. Under these FSMs is a table that presents the modes according to the states of the two FSMs product [2]. As represented in figure 2 we assume that the system can be in three modes: $Mod = \{r, s, s'\}$. Due to the principle of *one mode at a time* the set $S = \prod_{i=1}^{2} S_i$ of all states of the product $FSM_1 \times FSM_2$ is partitioned as:

$$S = \bigcup_{i \in r, s, s'} St(i),$$

where $St(i)$ is the set of system's states describing the mode i.

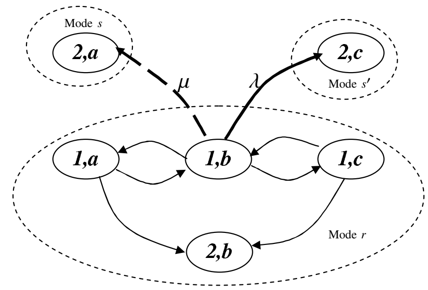


Fig. 2. Graph product and the three modes

Thus $St(r) = \{(1, a), (1, b), (1, c), (2, b)\}$; $St(s) = \{(2, a)\}$ and $St(s') = \{(2, c)\}$. One notices that $KeyCp\,(r, s) = \{FSM_1\}$ while $KeyCp\,(s, r) = \emptyset$. The mode s is thus permanent. Once the system reaches this function mode when the commutation event μ occurs, it can not leave it.

In general two diagnosis problems can be addressed. The first one is the scenario in which a component adopts a behavior without changing the system's global mode.The second scenario is the one in which a change in the state of a component is accompanied by a mode swishing. In the first case the tracking of the faulty component will be reduced to the set of components involved in the current mode (instead of all components in the system). We have proposed in [2] a monitoring and assessment approach using FSMs with special set of matrix operators to handle this problem. One can notice that, depending on the trajectory or the history of the system, it is possible to use different types of operators. Once the set of operators is determined and if a reference state is known it is easy to construct a FSM associated to the mode.
In the second case, assuming that we know the set $KeyCp(r, s)$ when the system switches from mode r to mode s, the tracking will take place only in this set. But in reality it is not easy to extract the set of key-components basing on minimal information.

2) The model: The FSM represented in figure 2 can be replaced by the automaton of modes described in figure 3.

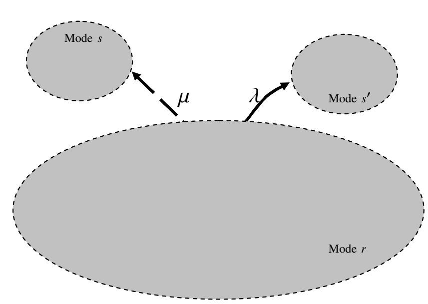


Fig. 3. Automaton of modes

In the present approach each mode k will be characterized by:

- the classe $St(k)$ of all involved states,
- and the set of its key-components $KeyCp(k)$.

With $St(k)$ it is possible to construct the set of all components acting in the current mode k (denoted $Unit(k)$).

Definition 3: A component U is a passive component of the mode k if its state in all global states in $St(k)$ remains unchanged and equal to an *idle state*.

Denoting by $PasCp(k)$ the set of passive components for the mode k, one has:

$$Unit(k) = Cp - PasCp(k).$$

It is obvious that $KeyCp(k) \subset Unit(k)$. That is each key-component of the mode k is obviously an active component of this mode.

If we denote by $\left\{U_i^{(k)};\ 1 \leq i \leq n\right\}$ the set $Unit(k)$, the FSM representing the Mode k is thus the graph product $\prod_{i=1}^{n} U_i^{(k)}$.

Using this FSM instead of the entire product (which includes passive components) can be a significant factor in reducing the size of the graph modeling the mode under study.

It seems evident that each mode must be known by its states by which it can be accessed (States-in) and the states in which it can be left (States-out). The set of States-in (resp. States-out) for a mode k will be denoted $InSt(k)$ (resp. $OutSt(k)$).

In the example of figure 2 $OutSt(r) = \{(1, b)\}$, $OutSt(s) = \emptyset$ and $IntSt(s) = \{(2, a)\}$.

The notion of States-out may be understanded as marked states commonly used in theory of control of discret event systems [4], [5]. The States-in may be viewed as initial states of the automaton representing the mode. For instance, after the system has completed its mission it reaches one of the states-out (or States-in) of the current mode, ready to perform a new task or to exit from this mode toward another State-in of a new mode, after a switching event (Figure 4).

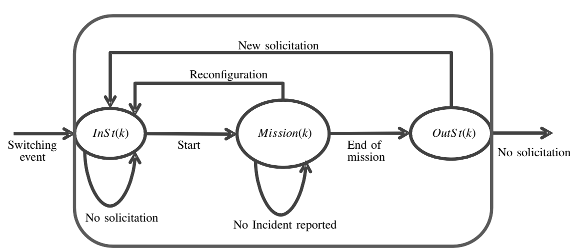


Fig. 4. Representation of the operation mode k

Obviously one has:

$$KeyCp(k) = \emptyset \Leftrightarrow OutSt(k) = \emptyset \quad \forall k \in Mod$$

In the sequel we denote by $Switch(k)^{\rightarrow}$ the set of all switching events to excite from mode k and $Switch(k)^{\leftarrow}$ the set of all switching events to enter this mode. The union of these two sets leads to the set of switching events denoted by:

$$Switch(k)^{\leftrightarrows} = Switch(k)^{\rightarrow} \cup Switch(k)^{\leftarrow}$$

At this stage, each mode will be represented by a quintuple as follows:

$$Mode\,k \equiv \left[Switch(k)^{\leftrightarrows},\ InSt(k),\ OutSt(k),\ KeyCp(k),\ Unit(k)\right]$$

From this model of the mode k, it is also possible to derive the submodels
$Mode\,k^{\rightarrow} \equiv [Switch(k)^{\rightarrow},\ OutSt(k),\ KeyCp(k),\ Unit(k)]$ and
$Mode\,k^{\leftarrow} \equiv [Switch(k)^{\leftarrow},\ InSt(k),\ KeyCp(k),\ Unit(k)]$. These models give all the information we need to study the system under a multi-model approach. They allow to have the whole view on each mode that can be visited by the system. The present approach is also adapted to the one presented in supervisory control of discrete-event systems discussed in [6], [4].

This model can be used to track component involved in a mode switching. But in this context the lake of knowledge about some of the sets mentioned in this model can limit this tracking.

III. Tracking of a component involved in modes' switching

The problem we address in this section is the tracking of the sets of components involved in the switch from $Mode\,r$ to $Mode\,s$, under the assumption that some of the set of the models are known. The tracking process follows the two steps hereafter:

First step: construction of the set $KeyCp(r, s)$ which includes all components that are susceptible to be involved in the switch $r \rightarrow s$,

Second step: identification of the components involved in this switch .

Depending on the knowledge that can be extracted from the system or the models we have, several cases can arise. The main challenge will be to have enough information about the different sets given by the models to compute the involvement of the components.

For example the set $Switch(r, s) = Switch(r)^{\rightarrow} \cap Switch(s)^{\leftarrow}$ gives the set of events on transitions responsible for the change $r \rightarrow s$. The simplest case is the one in which this set is reduced to one event. But it is also possible that the change $r \rightarrow s$ has more than one transition as in Fig. 5.

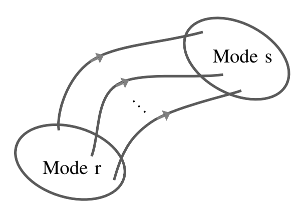


Fig. 5. Mode changing with several switching events

In the case showed in Fig. 5, one should proceed by thresholding of the level of involvement $Inv(i, \sigma)$ for all $\sigma \in$

$Switch(r, s)$. This method first consists in focusing on the set of components C_i for which $Inv(i, \sigma)$ is beyond a certain threshold $\tau < 1$. The case $0 < Inv(i, \sigma) < \tau$ will be considered only if the tracking process fails in the first case.

IV. Conclusion and further works

We have presented in this paper a framework for multi-model based approach for interconnected components network by finite state machine model. Our approach is based on the notions of key-component and level of involvement. These notions allow to propose a tracking of a component involved in modes' switching. Beyond the possibility provided by this approach for supervisory control of discrete event systems, one of its application is to add the controllers to the set of the system's component. Considering the system's controllers as part of the system opens on application of this approach to large class of systems.

References

[1] O. Kamach, L. Piétrac and É. Niel, *Multi-model approach for discrete event systems : application to operating mode management*, IMACS Multiconference Computational Engineering in Systems Applications, CESA'2003, Lille-FRANCE, 2003.

[2] B. Birregah, K. H. Adjallah, K. S. Assiamoua and P. K. Doh, *Grid systems monitoring and assessment using finite state machines with median symmetry operators*, IEEE International Conference on Systems Man and Cybernetics - ISIC, pp. 741-746, Oct. 2007

[3] R. Davis, *Diagnostic Reasoning Based on Structure and Behavior*, Artificial Intelligence, Vol. 24, 1984, pp. 347-410.

[4] G. Faraut, L. Pietrac and E. Niel, *Identification of incompatible states in mode switching*, IEEE International Conference on Emerging Technologies and Factory Automation - ETFA, pp. 121-128, Sept. 2008,

[5] S. Jiang and R. Kumar,*Decentralized control of discrete event systems with specializations to local control and concurrent systems*, IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol.30, no.5, pp.653-660, Oct. 2000.

[6] O. Kamach, L. Pitrac and E. Niel, *Multi-Model approach to discrete events systems: Application to operating mode management*, Mathematics and Computers in Simulation, Vol. 70, no.5-6, pp.394-407, 2006.

[7] M. Heymann, F. Lin and G. Meyer, *Multi-User Discrete Event Control with Active Events*, IEEE Transactions on Automatic Control, Vol. 47,no.2, pp. 314-318, Feb. 2002.

[8] Y. V. Kolokolov , A. V. Monovskaya, K. H. Adjallah, *Pulse Energy Conversion Systems: Real-time Emergency Forecasting*, To appear in IEEE Trans. on Energy Conversion