Learning Diagnoser and Supervision Pattern in Discrete Event System: Application to Crisis Management

Moussa Traore [1], Moamar Sayed-Mouchaweh [2], and Patrice Billaudel [3]

[1,3] *University of Reims, Champagne-Ardenne, CReSTIC, Reims, BP 1039, 51687, France*
moussa.traore@univ-reims.fr
patrice.billaudel@univ-reims.fr

[2] *University of Lille 1, F-59000 Lille, IA, Mines de Douai, 59500, France*
moamar.sayed-mouchaweh@mines-douai.fr

ABSTRACT

The increase of natural, industrial disasters and diverse crisis has stimulated more research interest in the world. A crisis can be industrial accident, train accident, earthquake, and etc. However, the crisis management is currently an important challenge for medical service and research, to develop new technical of decision support system to guide the decision makers. Crisis management is a special type of collaboration, therefore several aspects must be considered. The more important aspect or problem in a crisis management, is the coordination (and communication) between different actors and groups involved in the management. In this paper the focus is how to handle the coordination and interaction between these different actors and groups involved in crisis management by using a finite state automaton. The representation of the crisis management as a set of couple of states and events allows to optimize the crisis management by having real time the evolution of the situation and the prediction of their evolution at their earliest.

1. INTRODUCTION

Nowadays, there has been a lot of interest in crisis management. Because, in the last years, we assist to a growing number of disasters and diverse crisis, such as the Indian Ocean tsunami 2004, the Japanese earthquake and tsunami 2011, and ect (Reuter, Heger, & Pipek, 2013). Therefore, the response to these disasters and crisis (natural or man-made) have to be fast and effective. A fast and affective response in a crisis situation allows to reduce the disaster consequences on people and the damages in nearby areas. However, the response to a crisis situation requires the collaboration between different numerous people and groups, for example police, the personnel working in the site, first aid agents, Doctors, government delegates, position of victims. The figure 1 depict a given scenario for crisis management implying various actors and groups. When a disaster occurs, the people on the accident site send information to the communication center and the latter sends rescuer teams to the accident site or affected zone as shown in figure 1 (Benkhelifa, Moussaoui, & N-Taboudjemat, 2013). In crisis management several aspects must be considered. The more important aspect in crisis management, is the coordination between the actors and groups involved in the management. The coordination between different actors involved in crisis management is fundamental to reduce the disaster consequence on victims and nearest areas.

The new challenge of crisis situation is the representation of the crisis management as a set of couple of states and events to guide decision makers. In this paper, the coordination between different actors and groups involved in crisis management is viewed as discrete model (event). Discrete Event System (*DES*) are dynamic system whose the behavior is governed by occurrence of physical events that cause abrupt changes in the state of the system (Sayed-Mouchaweh & Billaudel, 2012). Most of the last approaches of *DES* is represented by Automaton (Yunxia, 2003; Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995; Kwong & Yonge-Mallo, 2011) and Petri Net (Cabasino, Giua, & Seatzu, 2010). The figure 2 illustrates an example of modeling of a crisis management as Finite State Automaton (FSA). The purpose of modeling the crisis management as FSA is to optimize the crisis management by having real time the evolution of the situation and the prediction of their evolution at their earliest. The FSA of the crisis management is used to generate languages (or sequence of events) for diagnosis purpose. The diagnosis in crisis management is not to detect failures, but to detect the critical situations. The critical situation in figure 2 is going twice in the state "x_5" during the crisis management. The state "x_5" corresponds to the waste of time of the intervention team to access to the crisis site, and in crisis management the waste time is unacceptable.

Recently, the prediction of *DES* based on the sequence of events (or equivalently, a trajectory) has increasingly interest of many researchers in the last years. The prediction, in general, is the process of making a decision about a future result or evolution of a situation. In literature, a lot of prediction approaches of *DES* are presented. The authors in (Genc & Lafortune, 2009) are presented a prediction method of a possibly unobservable event in the system behavior, based on the language containing the observable events. In (Takai & Kumar, 2012), the local prognosers of *DES* exchange their observations for the sake of arriving at the prognosis decision. The prediction problem in (Xi-Rien, 1989) is a special type of projection between two languages. We present in this paper, a new approach for the prediction of *DES* and adaptive detector of supervision pattern using *FSA*, which is based on the discrete model corresponding to the crisis management.

This paper is organize as follows. In section 2, we briefly review same notation and definition of the Automaton model of *DES*. In section 3, we describe the discrete model of crisis management. The standard diagnoser for the dynamic model is shown in section 5. A general definition of discrete event dynamic system is presented in section 4. We present in section 6, the prediction of *DES*. Finally, the learning diagnoser and adaptive supervision pattern are presented in section 7.

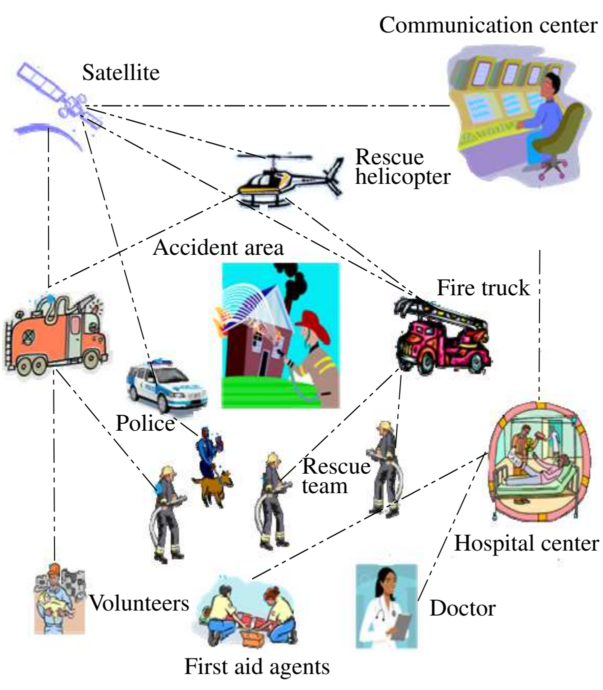

Figure 1. Emergency response scenario, (*I. Benkhelifa et al*).

2. Finite State Automaton

A Finite State Automaton can be defined as a six-tuple $M = (X, \Sigma, Y, \delta, x_0, F)$, where

- X is the set of states, Σ is the set of input events,
- Y is the nonempty finite set of outputs,
- $\delta : X \times \Sigma \longrightarrow X$ is the transition function,
- $x_0 \in X$ is the start (initial) state,
- $F \subseteq X$ is the (possibly empty) set of accepting or terminal states.

The finite set of events Σ can be partitioned in two subset, such that $\Sigma = \Sigma_o \cup \Sigma_{uo}$, where Σ_o is the observable events and Σ_{uo} is the unobservable events. A string is a finite-length sequence of events in Σ. The set of all strings formed by events in Σ is denoted by Σ^*. The set Σ^* is also called the Kleene-closure of Σ.

Further, we extend the transition function δ to $\hat{\delta}$ to accept words over Σ as following $\hat{\delta} : X \times \Sigma^* \longrightarrow X$. A state $x' \in X$ is reachable from the state x it there exists a sequence $T_M \in \Sigma^*$ such that $x' = \hat{\delta}(x, T_M)$ and we write $x \longmapsto x'$, and $x' = \hat{\delta}(x, T_M)$ is a path ξ in M if $x = x_0 \in X$. The state x_0 is called beginning of ξ and x_n is called the end of ξ. In the following, we call the strings $T_M = t_1 \cdots t_n$, with $t_1, \cdots, t_n \in \Sigma$ a trajectory in the system M. Thus, the path ξ is defined as $\xi = x_0 \xrightarrow{t_1} x_1 \cdots x_{n-1} \xrightarrow{t_n} x_n = x_0 \xrightarrow{T_M} x_n$.

Let T_M be a trajectory in Σ. For each trajectory $T_M \in \Sigma^*$, $|T_M|$ denoted its length. We say, the trajectory $T_M \in \Sigma^*$ is accepted by M if and only if there exists a path $\xi = x_0 \xrightarrow{T_M} x_n$, labeled by T_M, in the state diagram of M leading from start state x_0 to terminal state $x_n \in F$.

Any subset of Σ^* is called a language over Σ. The generated language of M, denoted by $\mathscr{L}(M)$ is defined as

$$\mathscr{L}(M) = \{T_M \in \Sigma^* \mid \hat{\delta}(x_0, T_M) \in X\}.$$

The language accepted by the system M is the set of all and only those trajectories over Σ that are accepted by M. The marked language accepted by M is defined by

$$\mathscr{L}_m(M) = \{T_M \in \Sigma^* \mid \hat{\delta}(x_0, T_M) \in F\}.$$

The language accepted by a deterministic *FSA* $\mathscr{L}_m(M)$ is called a regular language. A *FSA* of M is deterministic, if any given path in M labeled by trajectory $T_M \in \Sigma^*$ has a unique run, otherwise, *FSA* of M is non-deterministic.

The projection of strings from $\mathscr{L}(M) \longrightarrow \Sigma_o^*$ is denoted by $P : \mathscr{L}(M) \longrightarrow \Sigma_o^*$. Given a strings $T_M \in \mathscr{L}(M)$, P is obtained by removing all elements of Σ_{uo} in string T_M.

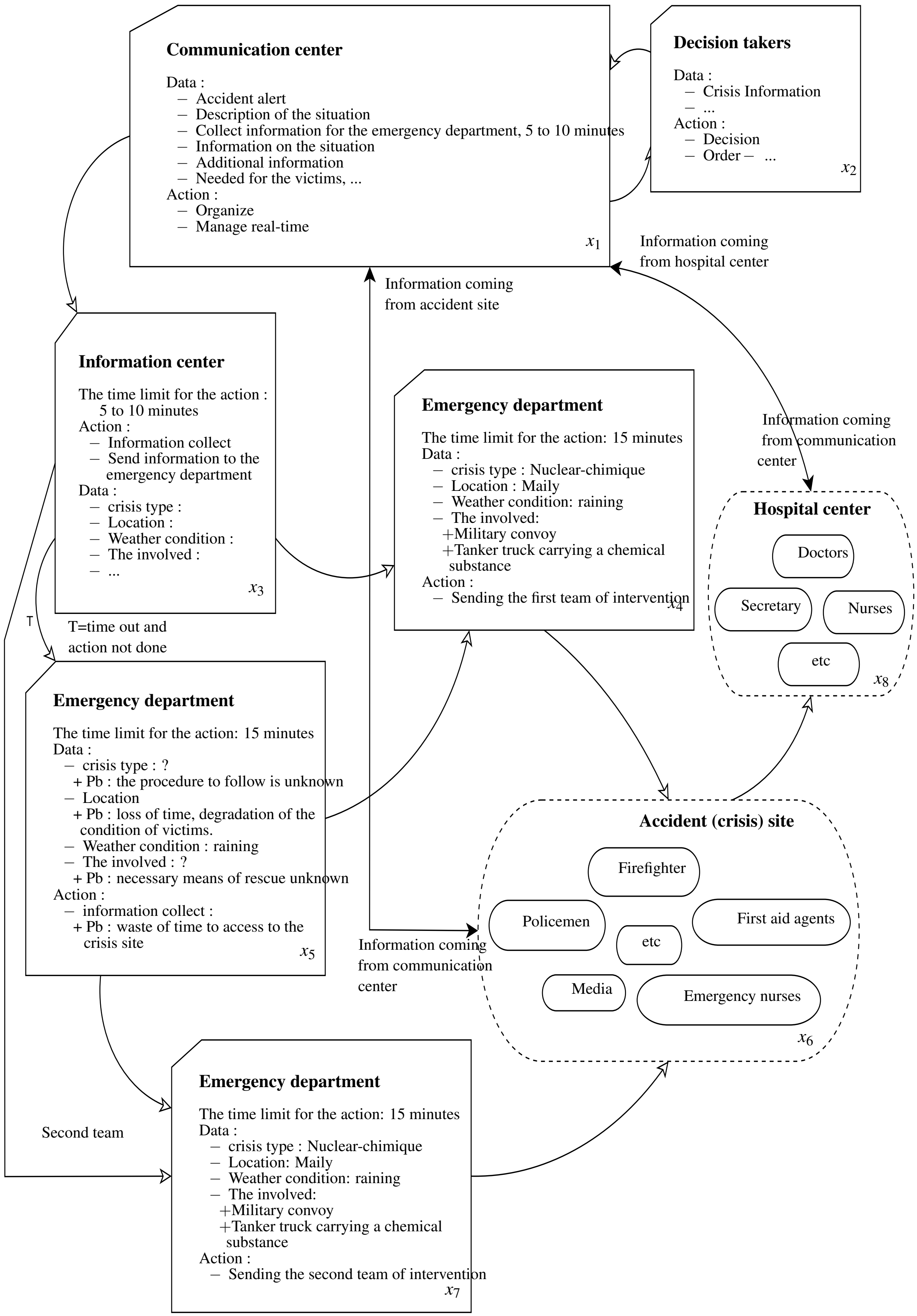

Figure 2. Interaction between different actors involved in the crisis management.

3. DISCRETE MODEL OF CRISIS MANAGEMENT

In this paper, the interaction between different actors and groups involved in crisis management is viewed as discrete model. The discrete model corresponding to the crisis management is modeled as a FSA. This is represented as a quintuple structure,

$$G = (X, \Sigma, Y, \varphi, x_0),$$

with φ is the transition relation, φ is the extension of δ of the system M, the relation φ has type $X \times \Sigma \rightarrow X \times Y$. For instance, $z' = (x', y') \in \varphi(x, t_i)$, with x, $x' \in X$, $y' \in Y$ and $t_i \in \Sigma$.

Example 1: *The Figure 3 shows the FSA of the crisis management corresponding to the figure2, with $x_0 = x_1$. In this example,*

- $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$,
- $\Sigma = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}\}$,
- $Y = \{y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8\}$,

The figure 3 shows a discrete event model and outputting y_i for all $i \in \{1, \cdots, 8\}$, when the system is in state x_i for $i \in \{1, \cdots, 8\}$, respectively.

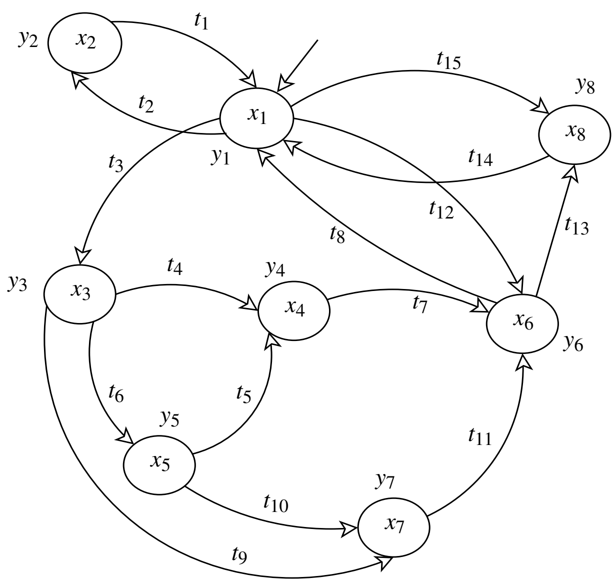

Figure 3. FSA model of the crisis management.

The FSA model of the crisis management shown in figure 3, allows one hand to monitor the communication and information between various groups involved in crisis management, and also to supervise same specific behaviors (or pattern) that can be a critical situation in the management. The notion of pattern means to define a language associated with a path of system G that we are interested in for the purpose of diagnosis. Other word, the pattern is defined as the recognition problem of the path whose intention is to answer the question whether trajectories corresponding to observed path are accepted or not by the model of the pattern. In (Ye & Dague, 2012), a pattern is define as a FSA. The language may be associated with the occurrence of single or multiple critical situation.

The transition function φ of $G = (X, \Sigma, Y, \varphi, x_1)$ can be extended to take input sequence. For example in figure 3, $\varphi(x_1, t_3) = \{x_3, y_3\}$ and $\varphi(x_1, t_3 t_9) = \{x_7, y_3 y_7\}$.

The equation $\varphi(x_1, t_3) = \{x_3, y_3\}$, means when the system G is in state x_1 and the communication event t_3 is emitted, the system G moves in the state x_3 and sends a communicate message y_3 (output).

Define two projections φ_1 and φ_2 of φ such that φ_1 gives the states reached from a state and an input given. The projection φ_2 defines the input/output pairs from state. These projections are defined as

$$\begin{cases} \varphi_1(x, t_i) = \{x' \in X \mid \exists\, y' \in Y \text{ such that } (x', y') \in \varphi(x, t_i)\}, \\ \varphi_2(x, t_i) = \{y' \in Y \mid \exists\, x' \in X \text{ such that } (x', y') \in \varphi(x, t_i)\}, \end{cases}$$

The projections φ_1 and φ_2 of φ may be extended as well to take input sequences. By applying φ_1 and φ_2 on the diagram represented in figure 3, we get

$$\varphi_1(x_1, t_3 t_9) = \{x_7\} \text{ and } \varphi_2(x_1, t_3 t_7) = \{y_3 y_7\}.$$

Let $L(G)$ be the language defined by the FSA G containing the input sequence allowed by G. Formally

$$L(G) = \{T_G \mid T_G \in \Sigma^* \text{ and } \Delta_G \in \varphi_2(x_1, T_G)\},$$

with x_1 start state and Δ_G output corresponding to the input T_G. The state $x \in X$ of G has an associated language

$$L_G(x) = \{T_G \mid T_G \in \Sigma^* \text{ and } \Delta_G \in \varphi_2(x, T_G)\},$$

with $\Delta_G = y_1 \cdots y_k$ and $T_G = t_1 \cdots t_k$ such that $y_1, \cdots, y_k \in Y$ and $t_1, \cdots, t_k \in \Sigma$. The language $L_G(x)$ is the set of all trajectory that originate from the state x of the system G. Clearly in figure 3, $L(G) = L_G(x_1)$.

Let $\mathcal{K}(L_G(x_1), t_i)$ be the trajectory that ends with t_i (Genc & Lafortune, 2009). Formally

$$\mathcal{K}(L_G(x_1), t_i) = \{T_G = T_0 t_i \mid T_0 \in \Sigma^* \text{ and } t_i \in \Sigma\}.$$

We recall here, the FSA model of a dynamic system is defined as $G = (X, \Sigma, Y, \varphi, x_1)$, where $\varphi : X \times \Sigma \longrightarrow X \times Y$ is the transition function.

4. DISCRETE EVENT DYNAMIC SYSTEM

In the literature, the event set Σ may include failure events $\Sigma_f = \{\Sigma_1, \cdots, \Sigma_p\}$. Indeed, a dynamic system can have p failure modes of critical situations $(F_1, \cdots, F_p)$ that describe the condition of the system. In addition to normal and failure modes the system may have a degraded mode, denoted N_d. Therefore, the state set X can be partitioned according to the condition of the system.

$$X = X_N \cup X_{N_d} \cup X_{F_1} \cup \cdots \cup X_{F_p}.$$

The condition of the system goes in X_{N_d}, when the degradation event occurs and in X_{F_i} when the failure event occurs.

To define the condition map of a dynamic system on a trajectory T_G of G, we introduce the label propagation function $LP : X \times \Omega \times \Sigma^* \to \Omega$. $LP(x, \lambda, T_G)$ propagates the label λ over $T_G \in \Sigma^*$, starting from $x \in X$ and following the dynamics of G, *i.e.* according $L_G(x)$, with $x \in X$, $\lambda \in \Omega$ and $L_G(x) \in \Sigma^*$ such that $T_G = \mathscr{K}(L_G(x), \mathscr{T})$.

$$LP(x,\lambda,T_G) = \begin{cases} N, & \text{if } \exists\, x' \in X \mid x' \in \varphi_1(x,T_G) \text{ and } x' \in X_N \\ N_d, & \text{if } \exists\, x' \in X \mid x' \in \varphi_1(x,T_G) \text{ and } x' \in X_d \\ F_i, & \text{if } \exists\, x' \in X \mid x' \in \varphi_1(x,T_G) \text{ and } x' \in X_{F_i} \end{cases}$$

The definition of the conditions map may be extended to subsets of X.

$$\text{for all } z \subseteq X,\ LP(z,\lambda_z,T_G) = \bigcup_{x_1 \xrightarrow{T_G} x_i \in z} \{LP(x_1,\lambda_i,T_G)\}.$$

Let $x_1, \cdots, x_m \in X$ and $m \in \mathbb{N}$ such that $z = \{(x_1,\lambda_1), \cdots, (x_m,\lambda_m)\}$. The system's condition λ_i is normal if $\lambda_i = N$ for all $1 \le i \le m$, certain if $\lambda_i = F_i$ for all $1 \le i \le m$ and uncertain if there exist $\lambda_j = N$ and $\lambda_i = F_i$ for same $1 \le i, j \le m$. Further detail about notions of certain and uncertain system's condition may be found in (Zad, Kwong, & Wonham, 2003) and (Genc & Lafortune, 2009).

Example 2: *Figure 3 shows a FSA model of a crisis management. We use the input of the system G to supervise the behavior corresponding to the critical situation. The critical situation that we want to detect is outputting twice y_5 during the crisis management. The first appearance of the output y_5 in the output sequence Δ_G brings the system into the set X_{N_d} corresponding to the degraded mode N_d. The second appearance of the output y_5 in the output sequence Δ_G during the crisis management brings the system into the set X_F corresponding to the critical mode. In this example*

$$X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$$
$$\Omega = \{N, N_d, F\}$$
$$Y = \{y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8\}$$

The necessary and sufficient condition for the pattern of a DES is based on the learning diagnoser and prediction of DES. The learning diagnoser is obtained from the standard diagnoser.

5. Standard diagnoser

A standard diagnoser denoted D_G must be able to detect and isolates faults and failures (Sampath et al., 1995), or to detect critical situations in crisis management. A fault implies a certain level of degradation of performance and a failure on the other hand denotes a complete operational breakdown of equipment or the process (Yunxia, 2003). A standard diagnoser is a FSA built for to detect and isolates faults and failures of G. Let $G = (X, \Sigma, Y, \varphi, x_0)$ be the discrete event model for the dynamic system that we want supervise. The set Y is the output of system G. The standard diagnoser that we use for discrete event dynamic systems is a FSA that takes the output sequence $\Delta_G = y_1 y_2 \cdots$ of system G as its input as shown in figure 4, with λ_i the condition functioning of the system.

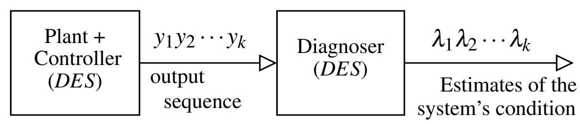

Figure 4. System and Supervision pattern

The standard diagnoser D_G of G is defined as $D_G = (Z, Y, \Omega, \zeta, z_0)$, with Z is the set of standard diagnoser state, Y is the set of standard diagnoser input, Ω is the set of standard diagnoser output, ζ is the standard diagnoser state transition function, the relation ζ has type $Z \times Y \to Z$, $z_0 \in Z$ is the start state of the standard diagnoser.

The diagnoser state space Z is the resulting subset of $2^{X \times \Omega}$ composed of the state of the diagnoser that are reachable from z_0 under ζ. The initial state z_0 of the diagnoser is defined by $z_0 = (x_0, \lambda_0)$. Assume the system G is normal to start, then $\lambda_0 = N$. State $z \in Z$ is given by

$$z = \{(x_1,\lambda_1),(x_2,\lambda_2),\cdots,(x_{n-1},\lambda_{n-1}),(x_n,\lambda_n)\},$$

where $x_i \in X$ and $\lambda_i \in \Omega$, for all $i \in \{1, \cdots, n\}$. In the following, $|z| = 1$.

Basing on the output sequence $\Delta_G = y_1 y_2 \cdots y_k$ of the system G, a state $z_k = (x_k, \lambda_k) \in Z$ is determined to which x_k may belong at the time that y_k was generated. For the diagnoser, the estimate of the system's condition from x_1 will be $LP(x_1, \lambda_k, \Delta_G)$ such that $(x_1, \lambda_1) \in \zeta(x_k, \lambda_k)$, with $z_1 = (x_1, \lambda_1)$ and $z_k = (x_k, \lambda_k)$.

The diagnoser state transition is defined by $z_{k+1} = \zeta(x_k, y_{k+1})$ with $z_k = (x_k, \lambda_k)$ and $y_{k+1} \in Y$. In the following, we write the diagnoser state $z_k = (x_k, \lambda_k)$ as $z_k = (x_{z,k}, \lambda_k)$. The standard diagnoser presented further above is shown in figure 5. Here, the pattern that has to supervise is having twice y_5 in the output sequence during the crisis management. *We remind that $y_1, y_2 \cdots \in Y$ are outputs of the system G and inputs for the diagnoser D_G.*

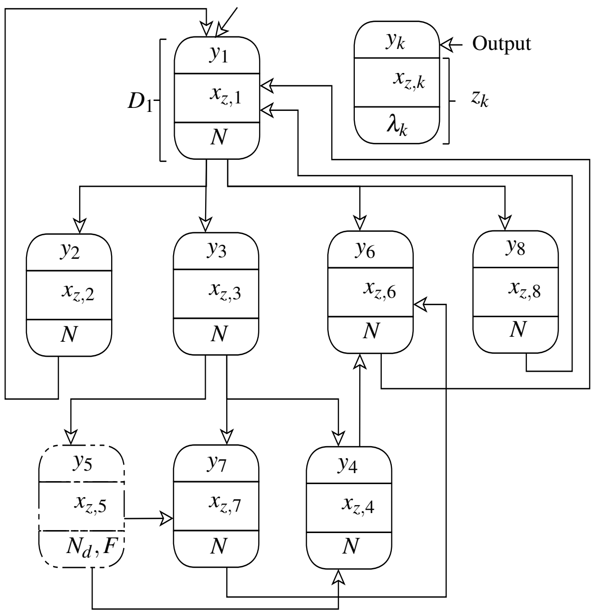

Figure 5. Standard diagnoser of the critical behavior in crisis management.

We address the problem of supervision pattern of a Discrete Event Dynamic System (*DEDS*). Let H be a bounded set of finite-length strings over Σ. The pattern can be define as a bounded H. The definition of the language that should be recognized by the supervision pattern depend the problem studied. In this paper, the critical situation (behavior) during the crisis management is detected if and only if.

$$\begin{cases} \mathscr{L}_{D_G}(x_{z,1}) = \{T_{D_G} \mid T_{D_G} \in Y^* \text{ and } \Delta_{D_G} \in \zeta_2(x_{z,1}, T_{D_G})\}, \\ \text{such that, it exists a language } L \in \Sigma^* \text{ defined by} \\ L = \{T_G \in \Sigma^* \mid T_{D_G} \in \varphi_2(x_1, T_G\} \text{ and } f(L) = true, \\ f(L) \text{ is a condition to define.} \end{cases}$$

In this paper, the condition is given as

$$\begin{cases} f(L) = |P_o(L)| \geq \mathrm{C}, \\ P_o : \Sigma^* \longrightarrow H^*, \ H \subseteq \Sigma, \\ \mathrm{C} = \text{Criteria, here C is a positive number,} \end{cases}$$

with P_o is the projection of strings, $T_{D_G} = y_1 y_1 \cdots$, and φ_2 is the extension of φ of G. For the behavior that we want to supervise here, we have $H = \{t_6\}$ and the Criteria $C = 2$. Until now, only the occurrence of t_6 brings G in the state x_5 as shown in figure 3.

The supervision pattern shown in figure 6 recognizes the language $\mathscr{L}_{D_G}(x_{z,1})$ if and only if the condition $|P_o(L)| \geq 2$ is verified.

The trajectory $\mathscr{L}_{D_G}(x_{z,k})$ is used to predict the evolution of the situation during the crisis management. The prediction,

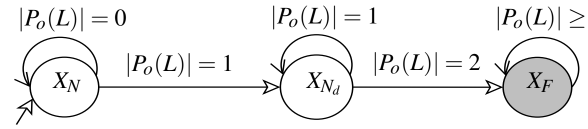

Figure 6. New supervision pattern for the critical situation in crisis management.

in general, is the process of making a decision about a future result or evolution of a situation. In the next section, we introduce the problem of prediction of discrete event system.

6. The prediction of *DEDS*

The prediction of a trajectory (or equivalently, sequence) of a dynamic system behavior is defined in the context of formal language.

Let $\mathscr{L}_{D_G}(x_{z,1})$ denote the set of all trajectory that originate from the start state diagnoser $z_1 = (x_{z,1}, \lambda_1)$, and $\mathscr{K}(\mathscr{L}_{D_G}(x_{z,1}, y_\alpha))$ the trajectory ends with $y_\alpha \in Y$.

$$\mathscr{K}(\mathscr{L}_{D_G}(x_{z,1}, y_\alpha)) = \{\beta \in Y^* \text{ such that } \beta = y_1 \cdots y_n y_\alpha\}.$$

Let ζ_1 and ζ_2 be the two projections of ζ of diagnoser D_G, with ζ_1 is given by

$$\zeta_1(x_{z,k-1}, y_k) = \{x_{z,k} \mid \exists\, \lambda \text{ such that } (x_{z,k}, \lambda) \in \zeta(x_{z,k-1}, y_k)\},$$

with $\lambda = LP(x_{z,1}, \lambda_k, \beta) \in \Omega$ if $y_\alpha = y_k$ and the state $z_k = (x_{z,k}, \lambda) \subseteq Z$ is the state estimate of D_G at time k,
and ζ_2 is defined by

$$\zeta_2(x_{z,k-1}, y_k) = \{\lambda \mid \exists\, x_{z,k} \in z_k \text{ such that } z_k \in \zeta(x_{z,k-1}, y_k)\}.$$

Let $\psi(x)$ be the function giving the state immediately after the state x. This function is defined as

$$\psi(x) = \{x' \mid \exists\, y \in Y \text{ such that } x' \in \zeta_1(x, y)\}.$$

Roughly speaking, a diagnoser state is predictable if it is always possible to detect the future diagnoser state, strictly before to arrive in this state. In this paper, we base only on the output sequence of *DEDS* model of system G to predict the future state or evolution. The prediction of the future diagnoser state at time k, when $x_{z,k}$ is generated, is given by

$$\widehat{x}_{z,k+1} = \psi(x_{z,k}) \cap \zeta_1(x_{z,k}, y_{k+1}),$$

with $y_{k+1} \in Y$ and y_{k+1} is the input of D_G.

The predicted state of the diagnoser D_G is :
$\widehat{z}_{k+1} = (\widehat{x}_{z,k+1}, y_{k+1})$.
Thus, the prediction of the trajectory $\widehat{\mathscr{K}}(\mathscr{L}_{D_G}(x_{z,1}, y_{k+1}))$ is the form: $\widehat{\mathscr{K}}(\mathscr{L}_{D_G}(x_{z,1}, y_{k+1})) = \{\widehat{\beta} = y_1 \cdots y_k y_{k+1}\}$.
The prediction of the system's condition is the propagation of the label λ_{k+1} over $\widehat{\beta}$, defined by $LP(x_{z,1}, \lambda_{k+1}, \widehat{\beta})$. Finally the diagnosis state predicted from $x_{z,1}$ is the form

$\widehat{z}_{k+1} = (\widehat{x}_{k+1}, LP(x_{z,1}, \lambda_{k+1}, \widehat{\beta}), x_k \longmapsto \widehat{x}_{k+1}$.

For instance, suppose at time k the output sequence $\Delta_G = y_1 y_3$ is observed, then the diagnoser state is $x_{z,k} = x_{z,3}$ and the system's condition from $x_{z,1}$ is $LP(x_{z,1}, \lambda_3, y_1 y_3) = N$.

When the system is in the state x_3 in figure 3, and if the next output symbol y_{k+1} is anything other than y_4, y_5, y_7, we get

$$\psi(x_{z,3}) \cap \zeta_1(x_{z,3}, y_{k+1}) = \emptyset,$$

that means the observation generated after y_3 is inconsistent with the model dynamic and the diagnoser cannot proceed. The currant diagnoser state $x_{z,k+1}$ is different to diagnoser state $\widehat{x}_{z,k+1}$ predicted before. Basing in the language $\mathscr{L}_G(x_1) = T_G$, in particular the output sequence Δ_G, we determine the state candidate.

When the output sequence is inconsistent with the model of the system G, then we have to revise the model of G by adding to its new transition that we believe are missing in the nominal model. This situation is not interpreted as a faulty situation, because we did not add new states. All the normal and fault modes are known and we add only the messing transitions. Adding new transitions in Σ of G is called learning diagnoser. In the next section we detail the construction of a learning diagnoser.

7. Learning Diagnoser

A learning diagnoser is a standard diagnosis that tolerant of missing transitions (information) about the system to be diagnosed. The learning diagnosis must be able to learn the true model of the system G, when missing information about the system are presented.

Let t_{new} be a new event detected and not found in Σ of system G. The new set of input events of G is given by $\Sigma_{new} = \Sigma \cup \{t_{new}\}$. A transition $x_d \xrightarrow{t_{new}} x_a$ is ordered pair of state denoting a transition from the state x_d to the state x_a. Let φ' be the extend function transition of φ of the system G such that

$$\varphi'_1(x, t_j) = \begin{cases} x_a \text{ if } x = x_d \text{ and } t_j = t_{new} \\ \varphi_1(x, t_j) \text{ otherwise} \end{cases}$$

Let be a dynamic model G' of G defines as $G' = extend(G, \Pi) = (X, \Sigma \cup \Pi, Y, \varphi', x_0)$. And G' is called the extension of G by Π, with Π is the set containing all the new transitions founded. The set transition Π is empty, if the model G of the system is consistent with the output sequence.

For instance in figure 2, when an accident happen, the information center is going to send data and actions to the emergency department. All the information are sent in 5 minutes, but with a wrong weather condition. For example, if the information center sends a temperature of 30°C in winter (Europa). Then the system is going to detect the temperature 30°C in winter in Europa is wrong. That mean the first team can not go before the correct temperature. During this wait, the first team loses time (waste of time) that mean in the diagram 3, the system is in the state x_5. The transition from the state x_4 to the state x_5 is a new transition for the system G. The resulting diagnoser, including the new transition, is shown in figure 7, then $\psi(x_{z,4}) = \{x_{z,6}, x_{z,5}\}$.

When the model of G is inconsistent with the output sequence, the subset H for the supervision pattern may be updated. In this paper, the critical situation that we want supervise is going twice in the state x_5, then with the new transition t_{new}, we can go to the x_5 (event t_6 or t_{new} occurs). Then the new subset for the supervision pattern is define by $H = H_{update} = \{t_6, t_{new}\}$. The critical pattern of behavior in crisis management is detected if and only if

$$\begin{cases} \mathscr{L}_{D_G}(x_{z,1}) = \{T_{D_G} \mid T_{D_G} \in Y^* \ \&\ \Delta_{D_G} \in \zeta_2(x_{z,1}, T_{D_G})\} \\ \text{such that it exist a language } L \text{ defined by} \\ L = \{T_G \in \Sigma^* \text{ such that } T_{D_G} \in \varphi_2(x_1, T_G\} \text{ and } |P_o(L)| \geq \text{C} \\ P_o^{new} : \Sigma_{new}^* \longrightarrow H_{update}^*, \ \Pi \supseteq H_{update} = H \cup \{t_{new}\} \\ \text{C = Criteria and in figure 6 } P_o = P_o^{new}. \end{cases}$$

$P_o^{new} : \Sigma_{new}^* \longrightarrow H_{update}^*$ is the new definition of P_o and Criteria=2. The fact to update the bounded set $H \subseteq \Sigma$, we obtain a learning supervision pattern (*see figure 6*).

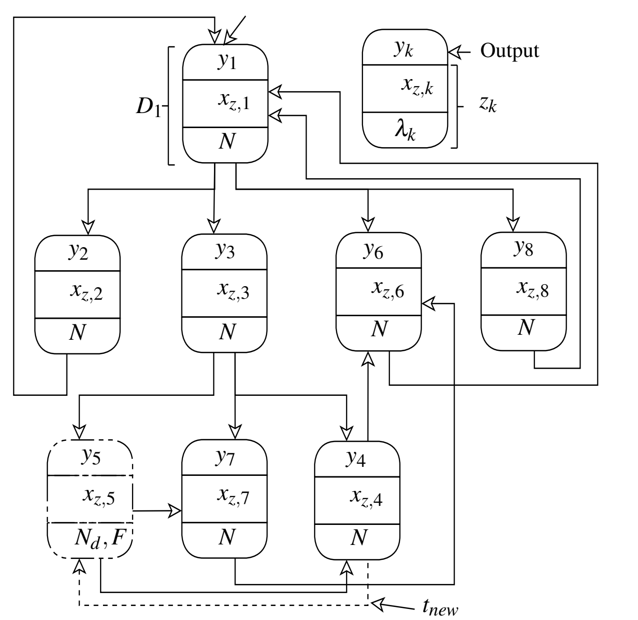

Figure 7. Learning diagnoser of the critical behavior in crisis management.

The diagnoser, as a Finite State Automaton, can be automatically translated into computer code. For example, the algorithm for a part of diagnoser 7, is given by

Algorithm 1 D_1

Require: $D_1 : z = (x_{z,k}, N)$ and y
  while $y == y_1$ **do**
    read y
    $z_1 \leftarrow (x_{z,k}, N)$
  end while
  if y==y2 **then**
    go to D_2
  else if $y == y_3$ **then**
    go to D_3
  else if $y == y_6$ **then**
    go to D_6
  else if $y == y_8$ **then**
    go to D_8
  else
    go to inconsistency
  end if

8. CONCLUSION AND PERSPECTIVES

In this paper, we proposed a representation of the crisis management as a set of couple of states and events. A learning diagnoser and prediction approaches are proposed and applied onto crisis management. Also, a method of adaptive supervision pattern is proposed in this paper.

Future work will also focus to introduction of the notion of the probability and to integrate time information onto the new transitions detected.

ACKNOWLEDGMENT

This work is supported by the CPER prject AidCrisis, sponsored by the Champagne-Ardenne region and the French ministry of higher education and research.

REFERENCES

Benkhelifa, I., Moussaoui, S., & N-Taboudjemat, N. (2013). Locating emergency responders using mobile wireless sensor networks. *In proceedings of IEEE, ACM, Proceeding ISCRAM, 10th Internationa Conference on Information Systems for Crisis Response and Management. Baden-Baden, Germany*.

Cabasino, M. P., Giua, A., & Seatzu, C. (2010). Fault detection for discrete event systems using petri nets with unobservable transition. *Automatica*, *Vol. 46, Issue 9*, 1531-1539.

Genc, S., & Lafortune, S. (2009). Predictability of event occurrences in partially-observed discrete-event systems. *Automatica*, *Vol. 45, Issue 2*, pp. 301 - 311.

Kwong, R., & Yonge-Mallo, D. (2011). Fault diagnosis in discrete-event systems: Incomplete models and learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *Volume 41 Issue : 1*, pp. 118-130.

Reuter, C., Heger, O., & Pipek, V. (2013). Combining real and virtual volunteers through social media. *In proceedings of IEEE, ACM, Proceeding ISCRAM, 10th Internationa Conference on Information Systems for Crisis Response and Management. Baden-Baden, Germany*.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995). Diagnosability of discrete event systems. *IEEE Transaction On Automatic Contol*, *vol. 40, No. 9*, 1555-1575.

Sayed-Mouchaweh, M., & Billaudel, P. (2012). Abrupt and drift-like fault diagnosis of concurent discrete event systems. *Machine Learning and Applications (ICMLA)*, *vol. 2*, 434 - 439.

Takai, S., & Kumar, R. (2012). Distributed failure prognosis of discrete event systems with bounded-delay communications. *IEEE Tansactions On Automatic Control*, *vol. 57, No. 5*, pp. 1259 - 1265.

Xi-Rien, C. (1989). The predictability of discrete event systems. *IEEE Transaction Automatic Contol*, *vol. 34 (11)*, pp. 1168-1171.

Ye, L., & Dague, P. (2012). A general algorithm for pattern diagnosability of distributed discrete event systems. *International Conference on Tools with Artificial Intelligence*.

Yunxia, X. (2003). *Integrated fault diagnosis scheme using finite state automaton* (Unpublished master's thesis). National University of Singapore.

Zad, S. H., Kwong, R. H., & Wonham, W. M. (2003). Fault diagnosis in discrete-event systems: framework and model reduction. *IEEE Transaction, on Automatic Contol*, *vol. 48, No. 7*, pp. 1199-1212.

BIOGRAPHIES

Moussa Traore is a Postdoctoral fellow at the University of Reims, France working under supervision of prof. M. Sayed-Mouchaweh and prof. P. Billaudel. He is working on Fault Diagnosis in Discrete Event System represented by Finite State Automaton. He is working on an AidCrisis project sponsored by Champagne-Ardenne region and the French ministry of higher education and research. He obtained his *Ph.D* degree (Control System and Signal Processing) from University Lille 1, France. He did his diploma in Safety Engineering (Optimization and safety of functioning Systems) from University of Technology of Troyes, France, 2006 and Bachelor's degree in Electrical Engineering from Faculty of Sciences and Technology of Nouakchott, Mauritania, 2004. His research work allowed him to publish 2 journal papers, 6 papers with proceedings at international conferences and 3 papers with proceedings at national (France) and international conferences (in French) in the area of diagnosis, prognosis and predictive maintenance.

Moamar Sayed-Mouchaweh received his Master degree from the University of Technology of Compiegne-France in

1999. Then, he received his PhD degree from the University of Reims-France in December 2002. He was nominated as Associated Professor in Computer Science, Control and Signal processing at the University of Reims-France in the Research center in Sciences and Technology of the Information and the Communication (CReSTIC). In December 2008, he obtained the Habilitation to Direct Researches (HDR) in Computer science, Control and Signal processing. Since September 2011, he is working as a Full Professor in the High National Engineering School of Mines Ecole Nationale Suprieure des Mines de Douai at the Department of Automatic Control and Computer Science (Informatique & Automatique IA). He supervised several defended Master and PhD thesis as well as research projects in the field of Modeling, Monitoring and Diagnosis in non-stationary environments. He published more than 100 journal and conference papers. He served as International Program Committee member for several International Conferences.