Adaptive Supervision of Patterns in Discrete Event Systems: Application to Crisis Management

M. Traore[1], M. Sayed-Mouchaweh[2], and P. Billaudel[3]

[1] *University of Technology of Troyes (UTT), Troyes, BP 2060, 10010, France*
moussa_amadou.traore@utt.fr
[2] *University of Lille 1, F-59000 Lille, IA, Mines de Douai, 59500, France*
moamar.sayed-mouchaweh@mines-douai.fr
[3] *University of Reims, Champagne-Ardenne, CReSTIC, Reims, BP 1039, 51687, France*
patrice.billaudel@univ-reims.fr

ABSTRACT

Crisis management is currently an important challenge for medical service and research. This motivates the development of new decision system approaches to assist (or to guide) the decision makers. A crisis management is a special type of collaboration involving several actors. The context and characteristics of crisis such as extent of actors and their roles make the crisis management more difficult in order to take decision. In this paper, we propose to model the interaction between different actors involved in crisis management. For this purpose we use finite state automaton in order to optimize the emergency response to the crisis and to reduce the disastrous consequences on people and environment. Thus, an adaptive supervision method is proposed. Therefore, we address the problem of diagnosis and prediction (prognostic) given an incomplete model of the discrete event systems of a crisis situation. When the model is incomplete, we introduce learning into the diagnoser (diagnosis module) construction.

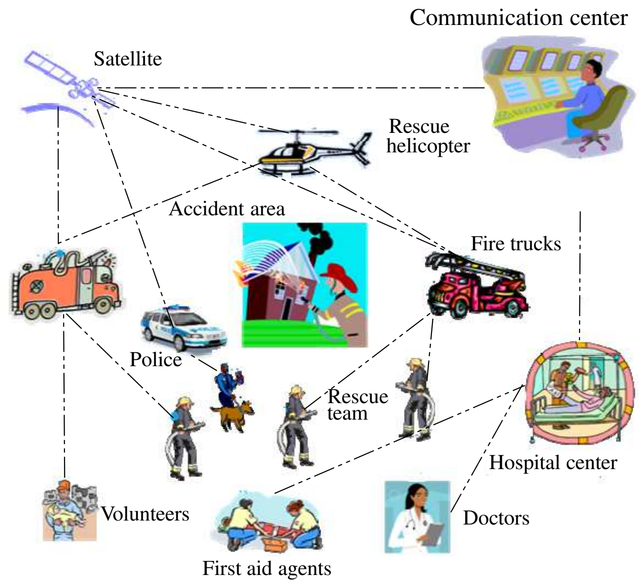


Figure 1. Emergency response scenario, (I. Benkhelifa et al).

1. INTRODUCTION

Nowadays, crisis management is an issue of paramount importance in the world (Fantacci, Marabissi, & Tarchi, 2010; Habib & Mazzenga, 2008). A crisis can be for instance an earthquake, an industrial accident, a train accident, etc. In general, the crisis management is a special type of collaboration involved several actors as policemen, first aid agents, doctors, government delegates, fire trucks, etc..., as shown in Figure 1. Further details about the collaboration among different actors involved in crisis management can be found in (Sediri, Matta, Loriette, & Hugerot, 2013).

During crisis management we face a problem of managing the people involved and the evolution of situations over time. Therefore several aspects must be considered. In particular, the coordination between different teams involved in the crisis management. This coordination between the teams is fundamental in order to optimize the emergency response and to reduce the disastrous consequences on people and the damages in nearby by surrounding areas. In this paper, the interaction between different actors and teams is represented by a discrete event model. The latter represents dynamic situation whose behavior is governed by the occurrence of physical events that causing abrupt changes in the state of the corresponding situation (Sayed-Mouchaweh & Billaudel, 2012). During crisis management, the cooperation implies the deployment of a rescue team, for example first aid agents, firefighter, etc $\cdots$ The execution and management this deployment is in general dynamic.

Recently, there has been a lot of interest in modeling crisis management or events in Discrete Event Systems (*DES*) to detect critical situations or to supervise a specific behavior. Different approaches have been developed to detect critical situations (or faults) or to supervise a specific behavior (supervision pattern) in the discrete event systems. Most of the latter approaches are represented by Finite State Automata (FSA) or Petri Net (Cabasino, Giua, & Seatzu, 2010; Sampath, Sengupta, Lafortune, Sinnamohideen, & Teneketzis, 1995; Yunxia, 2003). A supervision pattern has been proposed in (Jeron, Marchand, Pinchinat, & Cordier, 2006), but this approach requires a complete and accurate model of the system to be diagnosed. In this paper, we propose a new approach for supervision of pattern and prognostic of *DES* by using the inputs and outputs of the automaton. In this paper, the discrete event systems are modeled as FSA, which generates languages for the detection of critical situations during a crisis management or for supervisor control purpose. The languages generated by the automaton allow to build sequences of events represented by a string. Based on this formulation, a first approach of a learning diagnoser in discrete event systems is proposed in (Kwong & Yonge-Mallo, 2011). A learning diagnoser is a standard diagnosis that tolerate missing information (transitions) about the system (situation) to be diagnosed. The construction of a supervision pattern in discrete event systems using a sequence of events is described in detail in (Jeron et al., 2006).

The prognostic of the future evolution of discrete event systems based on trajectories has stimulated attracted a great deal of research interest in the last years. In this paper, prognostic aims to predict the critical situations of a discrete event before their occurrences (Khoumsi & Chakib, 2009). A prognosis framework in the case of a partially-observed discrete event systems is proposed in (Genc & Lafortune, August 2006). A so-called prognoser (prognosis module) issues a prognosis on whether a failure will occur, based on its partial observation of the plant. Also, a prognostic method of a possibly unobservable event in the system behavior, based on the language containing the observable events is presented in (Genc & Lafortune, 2009). In (Takai & Kumar, 2012), the local prognosers of discrete event systems exchange their observations for the sake of arriving at the prognosis decision. The prognostic problem in (Xi-Rien, 1989) is a special type of projection between two languages.

Crisis management is a special type of collaboration involving several different groups and actors. The challenge is how to handle the coordination and interactions between these different involved groups and actors during the crisis management and to detect abnormalities (e.g., critical process deviations, evolution towards dangerous or blocked situations, etc.) online or to predict the evolution of the current situation towards a dangerous or critical state. In this paper we developed a model based on finite state automata (called supervision patterns) describing the evolution of status of crisis management in response to actions and changes in its environment. The goal is to find supervision patterns leading to critical or dangerous situations or states. Then, a diagnostic model has been developed in order to recognize these special supervision patterns or even to predict their occurrence in order to alert the crisis decision makers of the evolution of crisis status in response to actions or made decisions. This alert will help the decision makers to adapt their actions (decisions) in order to stop the evolution of the crisis state towards critical or dangerous situations.

This paper is organized as follows. Section 2 introduces the concept and definition of discrete event systems. In section 3, we describe the definition of the supervision pattern. In section 4, the definition of discrete event systems model of a dynamic system is presented. The standard diagnoser for the dynamic system is shown in section 5. We present in section 6, the prognostic of discrete event systems. A learning diagnosis approach that tolerate missing transitions is presented in section 7. Finally, the approaches presented in this paper are illustrated to crisis management in section 8.

2. AUTOMATON MODEL FOR *DES*

A *DES* is a dynamic system that evolves in accordance with the abrupt occurrence, at possibly unknown irregular interval, of physical events. In this paper, the *DES* is modeled as FSA, which generates languages for the detection of critical situations or for supervisory control purpose.

A FSA G is a 6-tuple denoted as:

$$G = \{X, \Sigma, \delta, Y, x_0, F\},$$

where

- X is the set of fuzzy states
  $X = \{x_0, \cdots, x_i, \cdots, x_{n-1}, x_n\}$,
- Σ is set of input symbols,
  $\Sigma = \{\sigma_0, \sigma_1, \cdots, \sigma_{m-1}, \sigma_m\}$,
- The fuzzy subset $\delta : X \times \Sigma \to X$ is a function, called the fuzzy transition function. A transition from state x_i (current state) to x_j (next state) upon σ_k is denoted as: $x_j \in \delta(x_i, \sigma_k)$.
- Y is the non-empty finite set of output,
  $Y = \{y_0, y_1, \cdots, y_{l-1}, y_l\}$,
- $x_0 \in X$ is the set of initial fuzzy states and
- $F \subseteq X$ is the (possibly empty) set of accepting or terminal states. An example of FSA with accepting state is shown in figure 2, i.e., $F = \{x_1\}$.

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

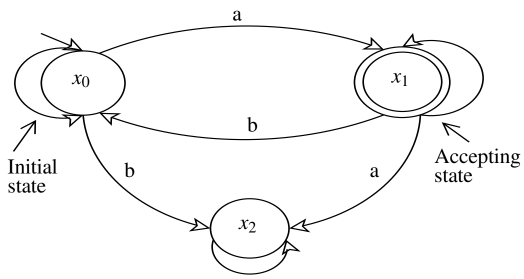


Figure 2. Example of an automaton for *DES*

the unobservable events. A string is a finite-length sequence of events over 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 $\widetilde{\delta}$ to accept words over Σ as following $\widetilde{\delta}: X \times \Sigma^* \longrightarrow X$. For example if $\delta(x_i, \sigma_i) = x_j$ and $\delta(x_j, \sigma_j) = x_k$, then $\widetilde{\delta}(x_i, \sigma_i\sigma_j) = x_k$ with $x_i, x_j, x_k \in X$ and $\sigma_i, \sigma_j \in \Sigma$.

Definition 1 *A state x_j is reachable from the state x_i if there exists a sequence $\Gamma \in \Sigma^*$ such that $x_j = \widetilde{\delta}(x_i, \Gamma)$. Also, we can say, the state x_j is the state-adjacent of the state x_i and we write $x_i \xrightarrow{\Gamma} x_j$.*

Let Γ_G be a trajectory in Σ. For each trajectory $\Gamma_G \in \Sigma^*$, $|\Gamma_G|$ denotes its length. We say, the trajectory $\Gamma_G \in \Sigma^*$ is accepted by G if and only if there exists a path ξ ($\xi = x_0 \xrightarrow{\Gamma_G} x_n$), labeled by Γ_G, in the state diagram of G leading from start state x_0 to terminal state $x_n \in F \subseteq X$. Any subset of Σ^* is called a language over Σ.

The language generated by FSA of G, denoted by $\mathscr{L}(G)$ is defined as

$$\mathscr{L}(G) = \{\Gamma_G \in \Sigma^* \mid \widetilde{\delta}(x_0, \Gamma_G) \in X\},$$

where $\widetilde{\delta}$ is the extension of δ that accept strings over Σ.

The language accepted by the system G is the set of all and only those trajectories Γ_G ($\Gamma_G \in \Sigma^*$) over Σ that are accepted by G. The marked language accepted by G is defined by

$$\mathscr{L}_m(G) = \{\Gamma_G \in \Sigma^* \mid \widetilde{\delta}(x_0, \Gamma_G) \in F\}.$$

Definition 2 *The language accepted by a deterministic FSA $\mathscr{L}_m(G)$ is called a regular language. A FSA G is deterministic, if any given path in G labeled by trajectory $\Gamma_G \in \Sigma^*$ has a unique run, otherwise, FSA G is non-deterministic.*

The FSA of G is said complete when all $(x, \sigma_i) \in X \times \Sigma$, $\mid \delta(x, \sigma_i)| \geq 1$ and a subset $X' \subseteq X$ is stable whenever that $\delta(X', \Sigma) \subseteq X'$.

The projection of strings from $\mathscr{L}(G) \longrightarrow \Sigma_o^*$ is denoted by P ($P : \mathscr{L}(G) \longrightarrow \Sigma_o^*$). Given a strings $\Gamma_G \in \mathscr{L}(G)$, P is obtained by removing all elements of Σ_{uo} in string Γ_G, (we recall $\Sigma = \Sigma_{uo} \cup \Sigma_o$), where Σ_o is the observable events and Σ_{uo} is the unobservable events.

In this paper, the language generated by FSA of G is used to detect critical situations or to supervise a specific behavior (pattern). The definition of a supervision pattern in discrete event systems is given in the next section.

3. DEFINITION OF THE SUPERVISION PATTERN

A supervision pattern is a language associated to a path of FSA M that we are interested in for the purpose of detection. The language may be associated with the occurrence of single critical situation or multiple critical situations. Thus, the language may be associated with a specific behavior of the situation or a system. In (Ye & Dague, 2012), a supervision pattern is a deterministic, complete FSA with a stable final states set F_ϑ. Let $\vartheta = (X_\vartheta, \Sigma_\vartheta, \delta_\vartheta, x_{0_\vartheta}, F_\vartheta)$ be the FSA which satisfies the four following conditions.

1. $\forall\, x \in X_\vartheta, \forall \sigma_j \in \Sigma_\vartheta$, if $x_1 \in \delta_\vartheta(x, \sigma_j)$ and $x_2 \in \delta_\vartheta(x, \sigma_j)$, then $x_1 = x_2$.
   - This condition describes the pattern as a deterministic FSA,
2. $\forall x \in X_\vartheta, \Sigma_\vartheta(x) = \Sigma_\vartheta$ where $\Sigma_\vartheta(x) = \{\sigma_j \in \Sigma_\vartheta \mid \exists x_1 \in X_\vartheta$ such that $x_1 \in \delta_\vartheta(x, \sigma_j)\}$.
   - This condition describes the pattern as a complete FSA,
   - A FSA M is said complete when all $(x, \sigma_i) \in X \times \Sigma$, $\mid \delta(x, \sigma_i)| \geq 1$ and a subset $X' \subseteq X$ are stable whenever $\delta(X', \Sigma) \subseteq X'$.
3. $F_\vartheta \subseteq X_\vartheta$ and $\delta_\vartheta(F_\vartheta, \Sigma_\vartheta) \subseteq F_\vartheta$, where $\delta_\vartheta(F_\vartheta, \Sigma_\vartheta) = \bigcup_{x \in F_\vartheta} \{x_1 \in X_\vartheta | x_1 \in \delta_\vartheta(x, \sigma_j)\}$ with $\sigma_j \in \Sigma_\vartheta$,
   - This condition characterizes that the final state set F_ϑ is stable.
4. $x_{0_\vartheta} \notin F_\vartheta$.

The supervision of the pattern ϑ 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 automaton ϑ. The supervision pattern method presented above is not adaptive for an incomplete model of a discrete event system. A discrete event model can arisen from abstraction and simplification of a continuous time system or through model building from input/output data. As such, it may not capture the dynamic behavior of the system completely. In next section, we present dynamic discrete event systems model in dynamic environment using the outputs sequences of M for the diagnosis. Most real-world applications operate in dynamic environment. In dynamic (non-stationary) environment, the parameters and structure of the application may change over time.

4. NEW AUTOMATON MODEL FOR *DES*

The pattern of the model of a system in non-stationary environment can be supervised by using the state and the output sequence. Suppose that X denotes the set of states and Y denotes the set of outputs. A sequence of state is a sequence $x_0 \cdots x_i$ of state and an sequence of output is a sequence $y_0 \cdots y_k$ for $x_0, \cdots, x_i \in X$ and $y_0, \cdots, y_k \in Y$. we introduce the same notations and definitions for the pattern of supervision of a behavior of a dynamic model. The behavior of a dynamic model to be supervised is modeled by applying methods based on FSA. These methods are represented by a quintuple structure

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

with φ is the transition relation, and φ is the extension of δ of the system M that is defined as $\varphi : X \times \Sigma \to X \times Y$.

Given state $x_i \in X$ and input (event) $\sigma_i, \in \Sigma$, and $z_j = (x_j, y_j) \in \varphi(x_i, \sigma_i)$ if and only if the input of σ_i when G is in state x_i may result in G in to state x_j and outputting y_j. The tuple defines by (x_i, z_j, σ_i) is a transition of the model G.

The equation $\varphi(x_0, \sigma_0) = \{x_1, y_1\}$, means when the system G is in state x_0 and the event σ_0 occurs, the system G moves to the state x_1 and sends the message y_1, with $x_0 \in X$, $\sigma_0 \in \Sigma$ and $y_0 \in Y$. The transition function φ of G can be extended to take the sequence of inputs. For example if we have $\varphi(x_0, \sigma_0) = \{x_1, y_1\}$ and $\varphi(x_1, \sigma_1) = \{x_2, y_2\}$, the extension of this example is $\varphi(x_0, \sigma_0\sigma_1) = \{x_2, y_1y_2\}$.

In this paper, we define two projections of φ. These two projections are φ_1 and φ_2. The projection φ_1 gives the states reached from a state and a given input. The projection φ_2 defines the output from state. These projections are defined as

$$\begin{cases} \varphi_1(x_i, \sigma_i) = \{x_j \in X \mid \exists\, y_j \in Y \text{ s.t } (x_j, y_j) \in \varphi(x_i, \sigma_i)\}, \\ \varphi_2(x_i, \sigma_i) = \{y_j \in Y \mid \exists\, x_j \in X \text{ s.t } (x_j, y_j) \in \varphi(x_i, \sigma_i)\}, \\ \varphi(x_i, \sigma_i) = (\varphi_1(x_i, \sigma_i), \varphi_2(x_i, \sigma_i)), \end{cases}$$

In the following $z_i = (x_i, y_i)$ and $z_j = (x_j, y_j)$.

The projections φ_1 and φ_2 of φ may be extended to take the input sequences. For $\varphi(x_0, \sigma_0) = \{x_1, y_1\}$ and $\varphi(x_1, \sigma_1) = \{x_2, y_2\}$, we get

$$\begin{cases} \varphi_1(x_0, \sigma_0\sigma_1) = \{x_2\}, \\ \varphi_2(x_0, \sigma_0\sigma_1) = \{y_1y_2\}, \end{cases}$$

Let $\mathscr{L}(G)$ be the language defined by the FSA G containing the input/output sequence allowed by G. Formally

$$\mathscr{L}(G) = \{\Gamma_G/\Delta_G \mid \Gamma_G \in \Sigma^* \ \& \ \Delta_G \in \varphi_2(x_0, \Gamma_G)\},$$

with x_0 the starting state. The state $x_i \in X$ of G has an associated language

$$\mathscr{L}_G(x_i) = \{\Gamma_G/\Delta_G \mid \Gamma_G \in \Sigma^* \ \& \ \Delta_G \in \varphi_2(x_i, \Gamma_G)\},$$

with $\Delta_G = y_0 \cdots y_k$, $\Gamma_G = \sigma_0 \cdots \sigma_k \in \Sigma^*$ and $y_0, \cdots, y_k \in Y$. The language $\mathscr{L}(G)$ is the set of all trajectories originating from the state x_0 of the system G. Clearly $\mathscr{L}(G) = \mathscr{L}_G(x_0)$.

Let $\Phi(L_G(x_0), \sigma_i)$ be the trajectory in $L_G(x_0) = \Gamma_G$ such that $\Gamma_G/\Delta_G \in \mathscr{L}_G(x_0)$ that ends with σ_i. Formally

$$\Phi(L_G(x_0), \sigma_i) = \{\Gamma_G = L_1\sigma_i \mid L_1 \in \Sigma^* \text{ and } \sigma_i \in \Sigma\}.$$

We recall here that the FSA model of a dynamic system is defined as $G = (X, \Sigma, Y, \varphi, x_0, F)$, where $\varphi : X \times \Sigma \longrightarrow X \times Y$ is the transition function. A set of events set Σ may include critical events (or faults). The event set of these critical events is denoted Σ_c. Thus, a dynamic system can have different functioning modes or situations: normal situation (N) and degraded situation (N_d). In addition to normal and degraded situations, there are p abnormal situations (failure modes), denoted $F_1, \cdots, F_p$ that describe the evolution of the situation in crisis management. The condition set of the dynamic system is defined as $\Omega := \{N, N_d, F_1, \cdots, F_p\}$. For a discrete event dynamic system, the state set X can be partitioned according to the condition of the system.

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

Occurrence of a degradation event brings the system into the set X_{N_d} corresponding to the degraded situation N_d. The occurrence of a critical event brings the system into the one of the set X_{F_i}, corresponding to the abnormal situation F_i.

To define the condition map of a dynamic system on a trajectory Γ_G of G, we introduce the label propagation function $L_\lambda : X \times \Omega \times \Sigma^* \to \Omega$. $L_\lambda(x, \lambda, \Gamma_G)$ propagates the label λ over $\Gamma_G \in \Sigma^*$, starting from $x_i \in X$ and following the dynamics of G, with $x_i \in X$, $\lambda \in \Omega$ and $L_G(x_i) \in \Sigma^*$ such that $\Gamma_G = \Phi(L_G(x_i), \ell)$.

$$L_\lambda(x, \lambda, \Gamma) = \begin{cases} N, \text{ if } \exists\, x_j \in X \mid x_j \in \varphi_1(x_i, \Gamma) \ \& \ x_j \in X_N, \\ N_d, \text{ if } \exists\, x_j \in X \mid x_j \in \varphi_1(x_i, \Gamma) \ \& \ x_j \in X_{N_d}, \\ F_i, \text{ if } \exists\, x_j \in X \mid x_j \in \varphi_1(x_i, \Gamma) \ \& \ x_j \in X_{F_i}, \end{cases}$$

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

$$\text{for all } z_k \subseteq X, \ L_\lambda(z_k, \lambda_{z_k}, \Gamma) = \bigcup_{x_0 \xrightarrow{\Gamma} x_i \in z_k} \{L_\lambda(x_1, \lambda_i, \Gamma)\}.$$

Let $x_0, \cdots, x_m \in X$ and $m \in \mathbb{N}$ such that $z_k = \{(x_0, \lambda_0), \cdots, (x_m, \lambda_m)\}$. The evolution of λ_i is normal if

$\lambda_i = N$ for all $0 \leq i \leq m$, certain if $\lambda_i = F_i$ for all $0 \leq i \leq m$ and uncertain if there exists $\lambda_j = N$ and $\lambda_i = F_i$ for the same $0 \leq i, j \leq m$.

The necessary and sufficient condition for the supervision pattern of a discrete event dynamic situation (or system) is based on the learning diagnoser and prognostic of discrete event dynamic system. The learning diagnoser is obtained from the standard diagnoser.

5. STANDARD DIAGNOSER

In this paper, a standard diagnoser, denoted here D_G is a FSA built to detect and isolate critical situations during crisis management whose the evolution is represented by the model G. This latter is defined by $G = (X, \Sigma, Y, \varphi, x_0)$. The model G is the discrete event model for the situation that we want to supervise. The set Y is the output of G. The standard diagnoser that we use for discrete event dynamic situations is a FSA that takes the outputs sequence $\Delta_G = y_0 y_1 \cdots y_k$ of system G as its inputs as shown in Figure 3, with λ_i the evolution of situations.

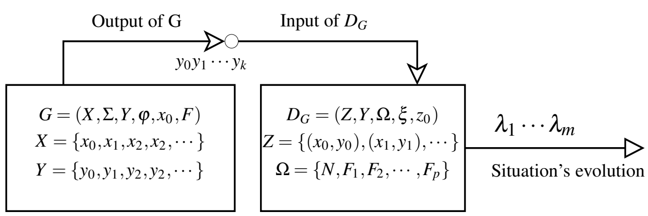


Figure 3. System and supervision pattern, with $\lambda_1 \cdots \lambda_m$ are the evolution of the situation and $y_0 y_1 \cdots y_k$ the output sequences.

The standard diagnoser D_G of G is defined as

$$D_G = (Z, Y, \Omega, \zeta, z_0),$$

where Z is the set of standard diagnoser state, Y is the set of standard diagnoser input, Ω is the set of standard diagnoser output (current state of the situation), ζ is the standard diagnoser state transition function, the relation ζ is $Z \times Y \rightarrow Z$, $z_0 \in Z$ is the start state of the standard diagnoser.

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

$$\begin{cases} \zeta_1(x_{i-1}, y_k) = \{x_i \mid \exists\, \lambda \text{ s.t } (x_i, \lambda) \in \zeta(x_{i-1}, y_k)\}, \\ \zeta_2(x_{i-1}, y_k) = \{\lambda \mid \exists\, x_i \text{ s.t } x_i \in \zeta(x_{i-1}, y_k)\}, \\ \zeta(x_{i-1}, y_k) = (\zeta_1(x_{i-1}, y_k), \zeta_2(x_{i-1}, y_k)). \end{cases}$$

with $\lambda = L_\lambda(x_0, \lambda_k, \beta) \in \Omega$.

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 that the system G is normal to start, then $\lambda_0 = N$. The state $z_j \in Z$ can be defined as

$$z_j = \{(x_0, \lambda_0), \cdots, (x_i, \lambda_i)\},$$

where $x_i \in X$ and $\lambda_i \in \Omega$, for all $i \in \{0, \cdots, n\}$. In the following we choose the length of equal 1, i.e., $|z| = 1$, $(eg z_k = \{(x_k, \lambda_k)\}$.

Based on the output sequence $\Delta_G = y_0 y_1 \cdots y_k$ of the model G, the state $z_k = (x_i, \lambda_i) \in Z$ is determined to which x_i may belong at the time that the output y_k was generated. For the diagnoser of the evolution of the situation from x_0 will be $L_\lambda(x_0, \lambda_i, \Delta_G)$ such that $\zeta((x_0, \lambda_0), y_0) \Rightarrow \zeta((x_i, \lambda_i), y_k)$.

For any $z_i = (x_i, \lambda_i)$ and $z_j = (x_j, \lambda_j)$, with $x_i,\ x_j \in X$ and $\lambda_i,\ \lambda_j \in \Omega$, we say that $z_j = (x_j, \lambda_j)$ is output-adjacent to $z_i = (x_i, \lambda_i)$ and we write $z_i \Rightarrow z_j$ if $\lambda_i \neq \lambda_j$ and if there exists $\tau \in \mathbb{N}$ and inputs $y_0, \cdots, y_\tau \in Y$ such that $(x_j, \lambda_j) \in \zeta((x_i, \lambda_i), y_0 \cdots y_\tau)$.

The diagnoser state transition is defined by

$$(x_i, \lambda_i) = \zeta((x_{i-1}), y_k) \ \text{ with } \ y_k \in Y.$$

In the following, we write the diagnoser state $z_k = (x_i, \lambda_i)$ as $z_k = (x_{z,i}, \lambda_i)$.

In this paper, we address the problem of supervision pattern of a discrete event model. We recall here, the supervision pattern means to define a language that we are interested in for the purpose of diagnosis and prognostic.

Let H be a subset over Σ, and $(H \subset \Sigma)$. The subset H is used to define the pattern that we want to supervise in the paper. The definition of the language that should be recognized by the supervision pattern depends the problem studied. In this paper, the detection approach of the critical behavior or situation is given by

$$\begin{cases} \mathscr{L}_{D_G}(x_{z,1}) = \{\Gamma_D / \Delta_D \mid \Gamma_D \in Y^* \ \& \ \Delta_D \in \zeta_2(x_{z,0}, \Gamma_D)\}, \\ \text{such that, } \exists\, L \in \Sigma^* \text{ defined by} \\ L = \{\Gamma_G \in \Sigma^* \ \mid \Gamma_D \in \varphi_2(x_0, \Gamma_G\} \ \text{ and } \ |P_o(L)| \geq \text{C}, \\ P_o : \Sigma^* \longrightarrow H^*, \ H \subseteq \Sigma, \\ \text{C = Criteria}, \\ \varphi_2 \text{ is the projection of } \varphi \text{ of the model } G, \end{cases}$$

with the sequence $\Gamma_D = \Gamma_{D_G} = y_0 y_1 \cdots$ that is the input sequence of the diagnoser D_G and the output of the model G .

Before to detect the critical behavior, it is interesting to predict the future state of the evolution of the situation by relying on the trajectory $L_{D_G} = \Gamma_{D_G}$. Prognostic of a discrete event systems aims at predicting failure events or critical situations of a discrete event systems before their occurrences and to detect new transitions that is missing in the nominal model. Each new transition must be validated by a crisis management expert after a predefined period of time (lifetime). If a new created transition is not validated by an expert during this time period, then the transition will be deleted. In the next section, we introduce the problem of prognostic of discrete event systems.

6. PROGNOSTIC OF DISCRETE EVENT DYNAMIC SYSTEM

Prognostic of a trajectory or equivalently, sequence of a dynamic system behavior is defined in the context of formal language.

Let $L_{D_G}(y_0)$ denote the set of all trajectories originating from the starting state of the diagnoser $z_0 = (x_{z,0}, \lambda_0)$, and $\Phi(L_{D_G}(y_0, y_\alpha))$ is a trajectory in $L_{D_G}(y_0)$ ending with $y_\alpha \in Y$.

$$\Phi(L_{D_G}(y_0, y_\alpha)) = \{\beta \in Y^* \text{ such that } \beta = y_0 \cdots y_n y_\alpha\}.$$

Let $\psi(x_{z,i})$ be the function giving the state immediately after the state $x_{z,i}$. This function is defined as

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

Roughly speaking, a diagnoser state is predictable if it is always possible to detect the future diagnoser state, immediately before to arrive in this state. In this paper, we rely only on the output sequence of discrete event dynamic system model of G to predict the future state.

The prognostic of the future diagnoser state at time k, when $x_{z,i}$ is generated, is given by

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

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

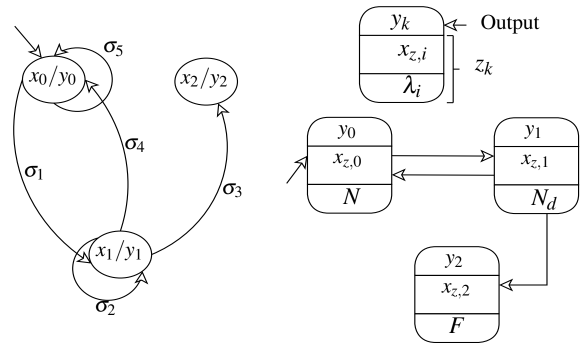
The predicted state of the diagnoser D_G is:
$\widehat{z}_{k+1} = (\widehat{x}_{z,i+1}, y_{k+1})$.

Thus, the prognostic of the trajectory $\widehat{\Phi}(L_{D_G}(y_0, y_{k+1}))$ is: $\widehat{\Phi}(L_{D_G}(y_0, y_{k+1})) = \{\widehat{\beta} = y_0 \cdots y_k y_{k+1}\}$.

The prognostic of the evolution of the situation is the propagation of the label λ_{k+1} over $\widehat{\beta}$, defined by $L_\lambda(x_{z,0}, \lambda_{k+1}, \widehat{\beta})$. Finally the diagnosis state predicted from $x_{z,0}$ in the form of $\widehat{z}_{k+1} = (\widehat{x}_{k+1}, L_\lambda(x_{z,1}, \lambda_{k+1}, \widehat{\beta}),\ x_i \longmapsto \widehat{x}_{k+1}$ (Traore, Sayed-Mouchaweh, & Billaudel, 2013).

For example, in this section we propose a FSA for the health condition of a patient as shown in Figure 4.(a). The states x_0, x_1 and x_2 are respectively excellent, poor and bad health condition of the patient.

- N: the patient's health condition is normal,
- N_d: the patient's health condition is in degraded state,
- F: the patient's health condition is abnormal,
- to characterize the active in example 4, we define $y_0 = [1\ \ 0\ \ 0]$, $y_1 = [0\ \ 1\ \ 0]$ or $y_2 = [0\ \ 0\ \ 1]$ respectively if the active state is x_0, x_1 or x_2 state.



(a) - model G (b) - diagnoser D_G

Figure 4. Finite state automaton corresponding to patient's health condition after a car accident, x_2 is the critical situation (condition), $\sigma_0, \cdots, \sigma_4$ are the drugs taken by the patient.

Suppose at time k the output sequence $\Delta_G = y_0 y_1 y_0$ is observed, then the diagnoser state is $x_{z,i} = x_{z,0}$. When the output sequence $\Delta_G = y_0 y_1 y_0$ is observed, and the next output symbol y_{k+1} is anything other than y_1, we get

$$\psi(x_{z,0}) \cap \zeta_1(x_{z,0}, y_{k+1}) = \emptyset,$$

that means the observation generated after y_0 is inconsistent with the model dynamic and the diagnoser can not proceed. The current diagnoser state $x_{z,i+1}$ is different to diagnoser state $\widehat{x}_{z,i+1}$ predicted before. Based on the language $\mathscr{L}_G(x_0) = \Gamma_G/\Delta_G$, in particular the output sequence Δ_G, we determine the sate candidate.

When the output sequence is inconsistent with the model of the situation G, then we have to revise the model of G by adding to it new transitions that we believe are missing in the nominal model. Adding new transitions in Σ of G is called learning diagnoser. In the next section we detail the construction of a learning diagnoser.

In non-stationary environments, the detection model must be updated in order to take into account the changes. The crisis environments are strong non-stationary environments.

7. LEARNING DIAGNOSER

A learning diagnoser is a standard diagnosis that tolerate 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 σ_{new} be a new input event not found in Σ of G and the new set Σ_{new} of G is given by $\Sigma_{new} = \Sigma \cup \{\sigma_{new}\}$. A transition $x_d \xrightarrow{\sigma_{new}} x_a$ is ordered pair of state denoting a transition from the state x_d to the state x_a. Let φ^{new} be extended transition function of φ of the system G such that

$$\varphi^{new}(x_d,\sigma_i) = \begin{cases} x_a \text{ if } \sigma_i = \sigma_{new} \ \& \begin{cases} \Sigma : \Sigma \leftarrow \sigma_{new}, \\ \text{and} \\ X : X \leftarrow x_a \quad \text{if} \quad x_a \notin X, \end{cases} \\ \varphi_1(x_d,\sigma_i) \ \text{otherwise,} \end{cases}$$

Let be a dynamic model G_{new} of the model G defines as

$$G_{new} = extend(G,\Pi) = (X, \Sigma \cup \Pi, Y, \varphi^{new}, x_0).$$

G_{new} is called the extension of G by Π, where Π is the set containing all the new transitions founded. The transition set Π is empty, if the model G of the system is consistent with the output sequence.

For instance, when the sequence $y_0y_1y_0y_2$ is observed by the diagnoser in Figure 4.(b), then the transition from the state $x_{z,0}$ to the state $x_{z,2}$ is a new transition for the model G, now $\psi(x_{z,0})$ can be

$$\psi(x_{z,0}) = \{x_{z,1}\} \ \text{ or } \ \psi(x_{z,0}) = \{x_{z,2}\}.$$

That means in Figure 4.(a), $x_2 \in \varphi_1(x_1,\sigma_2)$ and $x_2 \in \varphi_1(x_0,\sigma_{new})$.

When the model of G is inconsistent with the output sequence, the subset H for the supervision pattern may be updated, which is not the case for the supervision pattern approach presented in section 3.

When a new transition is detected, the detection model must be updated as

$$\begin{cases} \mathscr{L}_{D_G}(x_{z,0}) = \{\Gamma_{D_G}/\Delta_D \mid \Gamma_D \in Y^* \ \& \ \Delta_D \in \zeta_2(x_{z,0},\Gamma_D)\} \\ \text{such that it exist a language } L \text{ defined by} \\ L = \{\Gamma_G \in \Sigma^* \ \text{s.t} \ \Gamma_{D_G} \in \varphi_2(x_0,\Gamma_G\} \ \text{ and } \ |P_o(L)| \geq \text{C} \\ P_o^{new} : \Sigma_{new}^* \longrightarrow H_{update}^*, \\ \Pi = H_{update} = H \cup \{\sigma_{new}\}, \\ \text{C = Criteria and } P_o = P_o^{new}. \end{cases}$$

$P_o^{new} : \Sigma_{new}^* \longrightarrow H_{update}^*$ is the new definition of the projection P_o. The fact to update the subset $H \subseteq \Sigma$ in real time, we obtain a learning supervision pattern of the system (situation) operating in non-stationary environment.

8. APPLICATION TO CRISIS MANAGEMENT

During a crisis situation, the capacity to make fast and adequate decisions is a very important challenge for a better exit of crisis. The context and characteristics of crisis make more difficult to take decision than in normal situations. Thus, the multiplication of actors and roles in crisis management also increase the difficulty to exchange information and the coordination between different involved groups. That is why it is important to propose a model allowing to detect a critical situation during a crisis management, thus that the prognostic of the evolution of the normal situation toward this critical situation. The case of a critical situation, can be the management problem between different involved teams. In this paper we propose a model (no generic model) of crisis management applied on the team $S.A.M.U$ (Emergency Medical Service) from Hospital of Troyes in France, during $TEAN$ ($TEAN$ is the name of the exercise) exercise.

The Emergency Medical Service ($S.A.M.U$) is a hospital service which organizes emergency treatment outside the hospital (on the street, at home, etc). $S.A.M.U$ includes the center that receives calls made "15" (like 911 in the US) and is called specifically the Reception Center and Regulatory Appeals. It also includes an Academic Emergency Care Center. Mobile Service Emergency and Intensive Care include a medical team, vehicle and equipment responsible for responding to the request of the Emergency Medical Service.

$S.A.M.U.$ perform the following missions:

1. Ensure permanent Medical listening.
2. Determine and trigger, in the fastest time, the best-adapted response to the nature of the calls.
3. Ensure the availability of public or private hospital means adapted to the patient $(\cdots)$ and to prepare its welcome.
4. Otherwise, organize the transportation in a public or private institution involving a public service or a private company medical transport.
5. Ensure the patient's admission at the hospital by coordinating the hospital secretariat.

8.1. FSA model of $TEAN$ exercise

The $TEAN$ exercise team is composed of the following actors:

- Rear Base [1] (*RB*): Operations Coordination,
- Communication Center (*CC*): collecting information and sharing with *RB*,
- First Team: first intervention, sending the first evaluation (result) about the crisis to the *CC*,
- Advanced Medical Post (*AMP*): Intervention and evacuation of victims, sending the complete evaluation to the *CC*.

The FSA of the $TEAN$ exercise is shown in Figure 5.

The discrete event model showed in Figure 5 for $TEAN$ exercise, allows on one hand to monitor the communication and coordination between various groups involved in crisis management, and also to supervise some specific behaviors that are critical situations. The FSA in Figure 5 is specified as:

$$G_n = (X, \Sigma, \delta, Y, x_0, F), \ \ G_n \text{ is the nominal model}$$

[1] Other word, Rear Base is decision makers

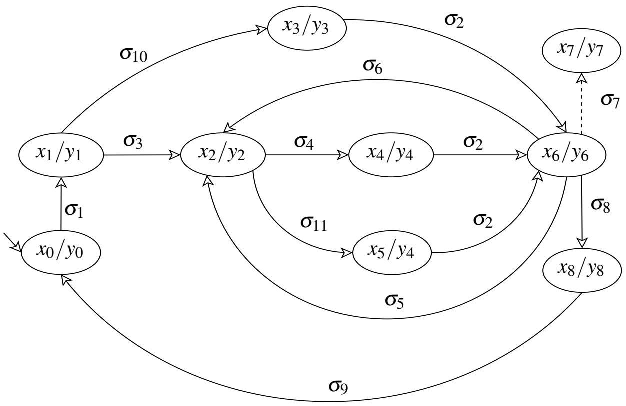


Figure 5. An example of modelisation of a scenario of crisis with finite state automaton.

The dashed line in Figure 5, between states x_6 and x_7 represents a critical event. The occurrence of the event $''\sigma_7''$ brings the system in the critical situation corresponding to state x_7.

In this example

$$X = \{x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}, \text{ is the set of states,}$$
$$\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \cdots, \sigma_{10}, \sigma_{11}\}, \text{ set of input symbols,}$$
$$Y = \{y_0, y_1, y_2, y_3, y_4, y_6, y_7, y_8\}, \text{ set of output events,}$$
$$F = \{x_7\}, \text{ terminal state,}$$

Table 1. List and definition of the states.

States	Definition
x_0	No crisis
x_1	Onset Crisis
x_2	Information received at the communication center (CC)
x_3	Information arrived at the police center
x_4	Information received at the Emergency department
x_5	Information arrived at the Advanced Medical Post (AMP)
x_6	Information received at the accident area
x_7	The model is not suitable for this crisis situation
x_8	The end of the intervention

Table 2. List and definition of the transitions (events).

events	Definition
σ_1	A call from (or about) an accident
σ_2	Sending Team to the accident site
σ_3	Sending information to CC
σ_4	Sending information to Emergency department
σ_5	Sending the first evaluation to CC
σ_6	Sending final evaluation to CC by the AMP's actors
σ_7	End of crisis management "without" success
σ_8	End of crisis management "with" success
σ_9	Confirmation of the end of the intervention
σ_{10}	Sending information to the police office
σ_{11}	Sending information to Emergency and AMP

Table 3. List and definition of outputs.

Output labels	Definition
y_0	No coming call
y_1	Accident is happen
y_2	Information arrived to CC
y_3	Information arrived to police office
y_4	Preparation of the Intervention Team
y_5	Preparation of the AMP
y_6	New Actors arrived in the accident area
y_7	uncontrolled situations (conditions)
y_8	The crisis is resolved

8.2. Diagnosis model of $TEAN$ exercise

The goal is to construct a diagnosis module (called diagnoser) able to diagnose critical situations. Hence, the standard diagnoser for the model illustrated in Figure 5 is shown in Figure 6, with $z_0 = \{x_0\}$. Each state of the diagnoser $D_{\widetilde{G}_n}$, shown as a rounded box in Figure 6 is a set of states of the system. In Figure 6, an output symbol ($\lambda(z_1)$ corresponding to the evolution of the situation is associated with each diagnoser state.

$$\lambda(z_i) = \begin{cases} F_1(\text{abnormal mode}), & \text{if i=7,} \\ N(\text{normal mode}), & otherwise. \end{cases}$$

Many critical situations can be identified in $TEAN$ exercise, but for simplicity and easy understanding of our approach, we cited only a single example of a critical situation.

Therefore, the specific behavior that we want to supervise (or to detect) here is the sequence $''y_6y_7''$ ($\Delta_{C_g} = y_6y_7$) in the output sequence during the crisis management. The appearance of Δ_{C_g} in the output sequence brings the crisis management into the set X_F corresponding to the critical situation F_1. Thus, the objective of the diagnoisis by a supervision pattern allows us to generalize the properties to be diagnosed.

The standard diagnoser D_{G_n} of the behavior presented just above is shown in Figure 6. This diagnoser is defined as $D_{G_n} = (Z, Y, \Omega, \zeta, z_0)$, with $\Omega = \{N, F_1\}$. The pattern that has supervised is having $''y_6y_7''$ in the output sequence during the crisis management. We remind that $y_0, y_1 \cdots \in Y$ are the outputs of the model G_n and inputs for the diagnoser D_{G_n}, as shown in Figure 3. In Figure 6, having the subset $''y_6y_7''$ in the output sequence brings the diagnoser in critical situation.

After the occurrence of the event σ_2 at time t, then, the prognostic of the future state at time $t+1$, when y_6 is generated, is given by

$$\widehat{x}_{z,i+1} = \psi(x_{z,6}) \cap \zeta_1(x_{z,6}, y_{k+1}).$$

The future state of the diagnoser D_{G_n} is:
$\widehat{z}_{k+1} = (\widehat{x}_{z,i+1}, y_{k+1})$.

Thus, the prognostic of the trajectory $\widehat{\Phi}(L_{D_{G_n}}(y_0, y_{k+1}))$ is: $\widehat{\Phi}(L_{D_{G_n}}(y_0, y_{k+1})) = \{\widehat{\beta} = y_0 \cdots y_6 y_{k+1}\}$.

The prognostic of the evolution of the situation is the propagation of the label λ_{k+1} over $\widehat{\beta}$, defined by $L_\lambda(x_{z,0}, \lambda_{k+1}, \widehat{\beta})$. Finally the diagnosis state predicted from $x_{z,0}$ is in the form of
$\widehat{z}_{k+1} = (\widehat{x}_{k+1}, L_\lambda(x_{z,1}, \lambda_{k+1}, \widehat{\beta}), x_i \longmapsto \widehat{x}_{k+1}$.

In this application:

$$\begin{aligned}
\widehat{x}_{z,i+1} &= \psi(x_{z,6}) \cap \zeta_1(x_{z,6}, y_{k+1}),\\
&= \{x_7, x_8\},\\
\widehat{z}_{k+1} &= (\widehat{x}_{z,7}, y_7),\\
&= (\widehat{x}_{z,8}, y_7),\\
\widehat{\Phi}(L_{D_{G_n}}(y_0, y_{k+1})) &= \{y_0y_1y_2y_3y_4y_5y_6y_7\} \implies \lambda_{k+1} = F_1,\\
&= \{y_0y_1y_2y_3y_4y_5y_6y_8\} \implies \lambda_{k+1} = N.
\end{aligned}$$

We address the problem of modeling the diagnosis objective by a supervision pattern of a discrete event dynamic model. The subset $H \subset \Sigma$ is used to define the pattern that we want to supervise in this application. In this paper, the critical behavior (situation) during the $TEAN$ exercise that we want to detect is detected if and only if

$$\begin{cases}
\mathscr{L}_\eta(x_{z,0}) = \{\Gamma_\eta/\Delta_\eta \mid \Gamma_\eta \in Y^* \text{ and } \Delta_\eta \in \zeta_2(x_{z,0}, \Gamma_\eta)\},\\
\text{such that, it exists a language } L \in \Sigma^* \text{ defined by}\\
L = \{\Gamma_{G_n} \in \Sigma^* \mid \Gamma_\eta \in \varphi_2(x_0, \Gamma_{G_n}\} \text{ and } |P_o(L)| \geq \text{C},\\
P_o : \Sigma^* \longrightarrow H^*, \text{ and } H = \{\sigma_7\} \subseteq \Sigma\\
\text{C = criteria},\\
\zeta_2 \text{ is the extension of } \zeta
\end{cases}$$

with $\eta = D_{G_n}$, $\Gamma_\eta = y_0 y_1 \cdots$ is the inputs and Δ_η is the outputs of the diagnoser D_{G_n} of the model G_n. For the behavior that we want to supervise here, we have $H = \{\sigma_7\}$ and the Criteria=1. The detection of σ_{new} between x_1 and x_7 and the occurrence of σ_7 and σ_{new} bring the crisis management in critical situation

If at time $t+1$, $\widehat{x}_{z,i+1} = \psi(x_{z,6}) \cap \zeta_1(x_{z,6}, y_{k+1}) = \emptyset$, that means the observation generated after y_6 is inconsistent with the model dynamic and the diagnoser can not proceed. Hence, we detect a new transition from x_6 to the next state at time $t+1$.

when the model of G_n is inconsistent with the output sequence, the subset H for the supervision pattern may be updated. In this section, the behavior that we want to supervise is having $''y_7''$ in the output sequence. For example if we get the output sequence $''y_1 y_7''$ at time $t+4$, then with the new transition σ_{new}, we can go to the x_7 when the event σ_{new} or σ_7 occurs. Then the new subset for the supervision pattern is defined by $H_{update} = \{\sigma_7, \sigma_{new}\}$. The critical pattern of the behavior of the crisis is detected if and only if

$$\begin{cases}
\mathscr{L}_\eta(x_{z,0}) = \{\Gamma_\eta/\Delta_\eta \mid \Gamma_\eta \in Y^* \text{ and } \Delta_\eta \in \zeta_2(x_{z,0}, \Gamma_\eta)\}\\
\text{such that it exists a language } L \text{ defined by}\\
L = \{\Gamma_{G_n} \in \Sigma^* \text{ such that } \Gamma_\eta \in \varphi_2(x_0, \Gamma_{G_n}\} \text{ and } |P_o(L)| \geq \text{C}\\
P_o^{new} : \Sigma^*_{new} \longrightarrow H^*_{update} = \Pi = H \cup \{\sigma_{new}\} = \{\sigma_6, \sigma_{new}\}\\
\text{C = Criteria and } P_o = P_o^{new}.
\end{cases}$$

$P_o^{new} : \Sigma^*_{new} \longrightarrow H^*_{update}$ is the new definition of P_o and C = 1.

Updating the subset $H \subseteq \Sigma$ allows us to obtain a learning supervision pattern.

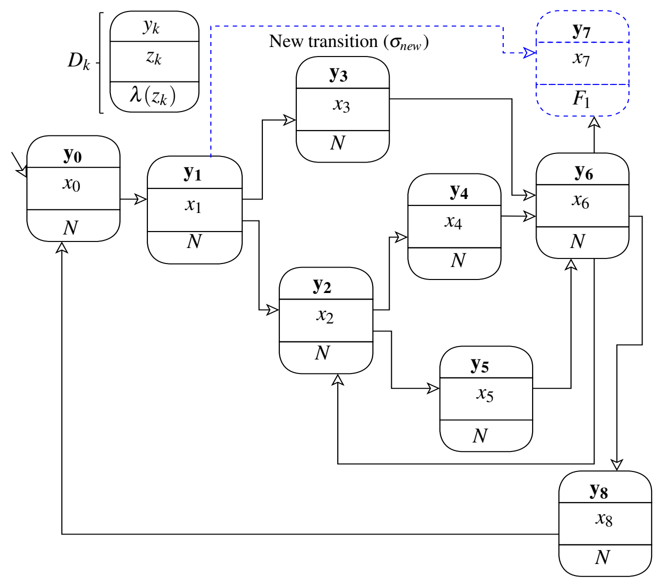


Figure 6. Diagnoser of crisp discrete event systems model shown in Figure 5.

9. NUMERICAL SIMULATION

In Figure 7, we show the numerical simulation of the crisis management of the $TEAN$ exercise model. For the simulation of the $TEAN$ exercise model , we used the Statechart simulation with Yakindu Statechart Tools. This latter is self contained and they not only contain the definition of states and state transitions, but also the definition of the statechart interface as shown in the left of the Figure 7. In Figure 7 the active state is "waiting" state. Therefore, the simulation that are made from the statecharts are complete and provide a well defined interface that can be easily integrated with any application code. The code generated by Yakindu statechart can be java, C or C++. The vector $LCD.outputSequence$ in Figure 7 is used for the diagnosis and prognosis of critical situation.

10. CONCLUSION AND PERSPECTIVES

In this paper, we showed that the interaction between different actors/groups involved in crisis management can be modeled

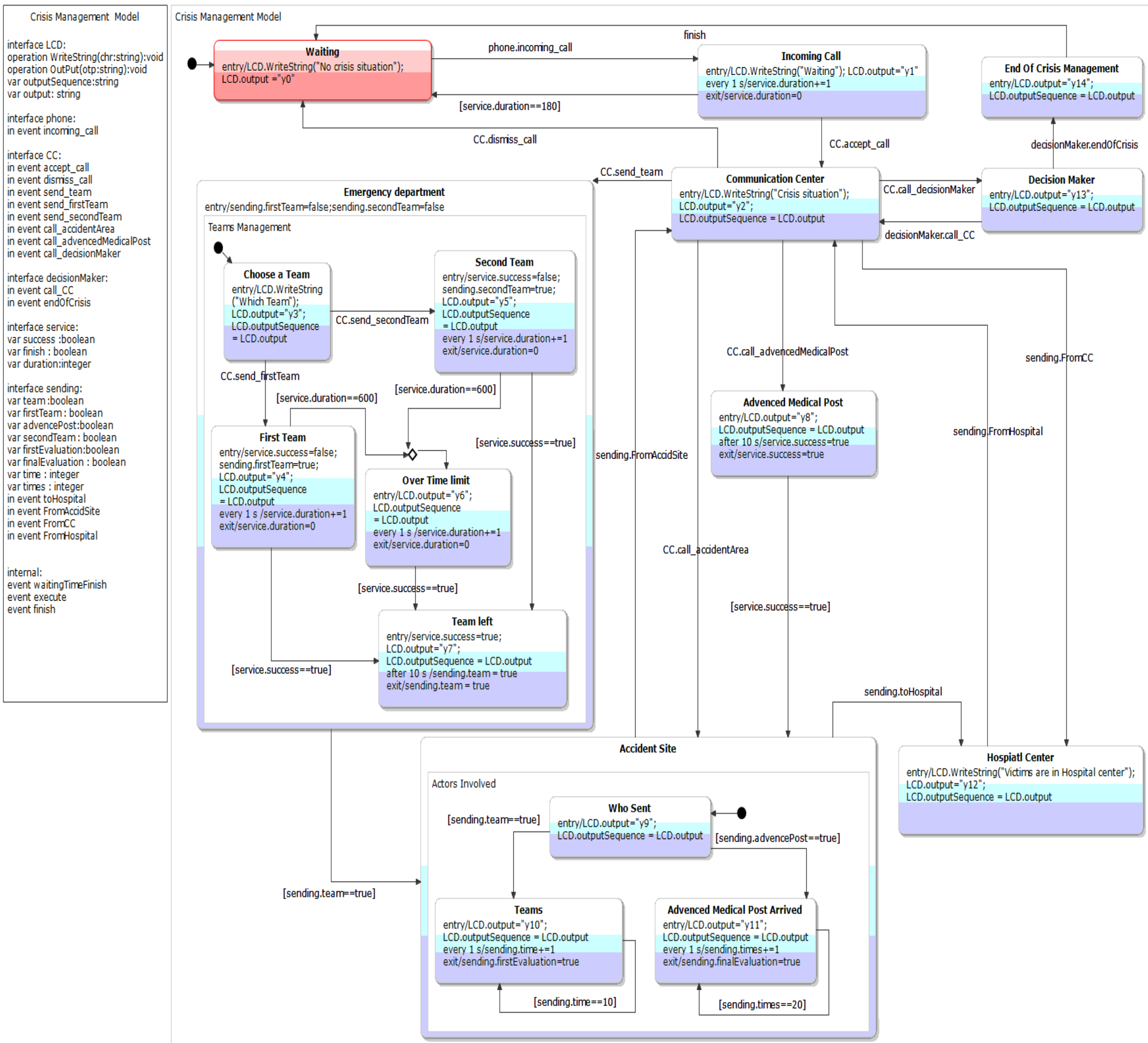


Figure 7. Crisis management model simulation of the *TEAN* exercise (this is not generic model).

by a FSA to help the decision makers to act during the crisis. The notion of supervision pattern for the diagnosis purpose is presented and illustrated with a crisis management as a case study. A method for the prediction of discrete event systems model of a dynamic system is presented. We we also presented a learning diagnoser that is tolerant to missing transitions (information) about the system to be diagnosed. We demonstrated how the learning diagnosis learn the true model of the system. We proposed a new learning supervision pattern for discrete event dynamic systems applied to a crisis management case.

Future work will focus on the development of a generic learning supervision pattern for the dynamic model of a crisis. We will propose a new diagnosis and prognosis approaches that deal with the problem of fuzziness, impreciseness and uncertainness, like stress of the people involved in crisis management and the weather condition.

ACKNOWLEDGMENT

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

REFERENCES

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

Fantacci, R., Marabissi, D., & Tarchi, D. (2010). A novel communication infrastructure for emergency management: the in.sy.eme. vision. *Wireless Communications & Mobile Computing*, *10 (12)*, pp. 1672-1681.

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

Genc, S., & Lafortune, S. (August 2006). Predictability in discrete-event systems under partial observation. *In 6 th IFAC Symposium on Fault Detection, Supervision, and Safety of Technical Processes, Beijing, PR China,.*

Habib, I., & Mazzenga, F. (2008). Wireless technologie advances for emergency and rural communications. *IEE Wireless communications Magazine*, *15 (3)*, 6-7.

Jeron, T., Marchand, H., Pinchinat, S., & Cordier, M.-O. (2006). Supervision patterns in discrete event systems diagnosis. *Workshop on Discrete Event Systems, WODES'06, Ann-Arbor (MI, USA).*

Khoumsi, A., & Chakib, H. (2009). Multi-decision decentralized prognosis of failures in discrete event systems. *American Control Conference*, pp. 4974-4981.

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*, *41 (1)*, pp. 118-130.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., & Teneketzis, D. (1995). Diagnosability of discrete event systems. *IEEE Transaction On Automatic Contol*, *40 (9)*, pp. 1555-1575.

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

Sediri, M., Matta, N., Loriette, S., & Hugerot, A. (2013). Crisis clever, a system for supporting crisis managers. *In proceedings of IEEE, ACM, Proceeding ISCRAM, 10th Internationa Conference on Information Systems for Crisis Response and Management. Baden-Baden, Germany.*

Takai, S., & Kumar, R. (2012). Distributed failure prognosis of discrete event systems with bounded-delay communications. *IEEE Tansactions On Automatic Control*, *57 (5)*, pp. 1259 - 1265.

Traore, M., Sayed-Mouchaweh, M., & Billaudel, P. (2013). Learning diagnoser and supervision pattern in discrete event system: application to crisis management. *Conference of the Prognostics and Health Management Society, New Orleans, USA*, pp. 694-701.

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.

BIOGRAPHIES

Moussa Traore is a Postdoctoral fellow at the University of Technology of Troyes, France working under supervision of prof. Eric Chatelet. He is working on Fault Diagnosis in Discrete Event Systems represented by Finite State Automaton. He is working on an 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 received his Master of Science (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 area is diagnostic and prognostic of dynamic systems in non-stationary environment, for continuous and discrete systems. 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 defendedMaster 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.

Patrice Billaudel Patrice Billaudel obtained his PhD degree in 1990 and became Associated Professor in Automation at the Higher School of Technical Training of the University of Reims Champagne-Ardenne located in Charleville Mzires. He is Full Professor there since he obtained his Accreditation to Direct Research in 1999. He is in charge of the Masters Degree in Material Science and New Technologies and he was head of the school during 3 years. He is working in the Research Center in Science and of Information and Communication Technologies (CReSTIC). He supervised Master and PhD theses in the field of biomedical sensors, signal processing, monitoring and diagnosis. He published about 100 journal and conference papers and he is a reviewer for journals and conferences.