Determining of critical and dreaded states achieved during metro line supervision

Delphine Paquereau[1,2], Laurent Piétrac[1], Eric Niel[1] and Laurent Bouresche[2]

Abstract— **The research developed here comes within a global approach of a metro line supervision study. Methodology of supervisory control theory is applied to avoid undesirable behaviors during incident situations management and so enforce passengers safety. In this paper, unlike the forbidden state problem, the set of states not to reach is not a given parameter. This set corresponds to safety and controllability constraints: the sets of critical and dreaded states are defined and determined. An example is presented in transport systems area and the developed algorithm is used to identify potentially unsafe situations which do not ensure passengers safety.**

Index Terms— **Supervision, Transport system, Process control, Petri nets, Safety**

I. Introduction

Nowadays, human safety is essential in industrial systems, for example in nuclear area, air transport or chemical industries. However, in complex systems, having the complete mastery is impossible, incident may occur in a uncontrollable way and unsafe situations exist. A total safety, forbidding unsafe situations, could amount to a not available system. To manage unsafe situations, procedures detail actions to realize to ensure the people safety for the best.

For years, in a metro network, the number of passengers has been growing up regularly. Moreover, the more there are trains on a line, the more risks of incidents increase. Therefore, metro line supervision has been changing: human operator role has been modified by monitoring and control computer systems. When an incident occurs (an event which may cause people or material damage), procedures exist to manage it and then protect passengers and provide transport service as soon as possible [2].

In metro line supervision context, the study objective is to analyze incidents management procedures with the aim of improving people safety. This is based on system modeling and on valuation of safety. Included in a global approach, this paper is focused on this identification and calculation of the unsafe situations as well as the way to avoid them.

The paper is structured as follows. Section II introduces metro line supervision, more specifically in an incident context. The successive steps of the study are detailed in section III to give an overview of the research works. Supervisory control theory is presented in Section IV and a state of the art of control system with forbidden states. Section V introduces notations and definitions of the studied sets and develops an algorithm to determine dreaded and critical states. Section VI shows an application of the algorithm in a transport system.

II. Metro line supervision

Metro line supervision is realized by an operator, called a dispatcher, who is responsible of trains traffic management and passengers safety. He deals with one metro line from a unique place called Operation Control Center (OCC).

A. *Traffic management*

Supervision system, Automatic Train Supervision (ATS) [1], manages the traffic on modernized line. ATS provides in real-time transport supply monitoring by overseeing signalling system (the acquisition and control systems directly connected to the ground).

ATS software developed by Thales allows the dispatcher to oversee that transport supply progresses according to expected operating program. Thus, the dispatcher knows in real-time notably all trains position, the line power supply and the drivers availability. Trains punctuality and frequency are also managed to offer the best transport service for passengers. These features are automatic and ensure a satisfactory traffic management. All the traffic and punctuality functionalities are not studied in this paper because they do not really need improvements.

B. *Safety management*

When an incident occurs, the dispatcher has to gain control of operating and remains the only one decision-maker for undertaking emergency measures. He is in a stressful context since he is in charge of people safety. His responsibilities are important and intensified by constraints like the number and the diversity (passenger to fireman) of people present on the metro line.

The dispatcher may communicate with all drivers on the line to have a better outline of the situation. He also may adapt the operating with deleting trains or changing their journey on the line. Then, transport conditions are sorely damaged, the operating is in a degraded mode and the dispatcher sets up elaborated strategies, according to his experience and knowledge. He has to go back to a usual operating [13] and to strive to reduce the inconvenience caused by incident on passengers.

C. *Incident context*

When trains are running, in operating day, many incidents can disrupt the proper traffic and thus decrease considerably

[1]Laboratoire AMPERE UMR 5005 - INSA Lyon - Bâtiment Saint Exupéry, 25 Avenue Jean Capelle, 69621 Villeurbanne, France.
`first name.name@insa-lyon.fr`

[2]THALES COMMUNICATIONS & SECURITY,20 Rue Grange Dame Rose, 78140 Vélizy, France.
`first name.name@thalesgroup.com`

transport service for passengers. These incidents are classified in two ways: depending on their sources and on their impact on traffic. Elements which may cause an incident are classified in five categories: line equipment, trains, power supply equipment, passengers and external environment.

These incidents are also arranged depending on their effect on traffic. An incident is minor if its impact on traffic remains limited and is managed automatically, without requiring any dispatcher actions. For example, when numerous people want to get on train, it is possible that the train stays longer at station and then is later than scheduled. Upstream and downstream trains advance is modified automatically to reduce the distance between trains. An incident is more serious if the repercussions on passengers safety and line operating are more significant, in this case, the dispatcher intervention is essential. A signalling failure, like a traffic lights malfunction, may induce important lateness. So, the dispatcher has to ask somebody to go and fix the faulty equipment. Only this kind of serious incidents is studied later.

To manage an incident and provide solutions to resume train traffic, each transport company, Thales ATS user, devises its own procedures. A procedure describes step by step actions to realize by the dispatcher, needed authorizations to go to the next step and communications between drivers and dispatcher. In order to react as soon as possible to incidents which affect passengers safety, procedures are memorized by dispatchers but implemented according to their expertise. Daily used on metro line, these procedures are safe by usage. There is no blockages when they are executed, a solution always exists to end procedures.

A description of different sorts of incidents and a classification and formalization of Paris metro company (RATP) procedures have been presented in a previous paper [12].

A procedure manages generally only one incident [13] and does not take simultaneous occurrence of several incidents into account. There is no coordination between existing procedures. Moreover, it is not possible to prevent incidents from occurring. When one of them occurs, a danger appears and may not be avoided. Thanks to these procedures, the danger for passengers is brought under control but does not disappear. For example, when a fire breaks out, nothing may prevent it to occur and passengers are in an unsafe situation because of smoke in the tunnel. So the dispatcher realizes actions to protect passengers, like an evacuation. In the same time, firemen try to control the fire but the danger does not necessarily disappear immediately for passengers.

III. Global approach

In this paper, the first objective is to identify the unsafe situations which may be reached during the management of several incidents in the same time. The second objective is a control objective: reacting in a controllable way to go to a safer situation and avoiding situations where the only possibility is to wait for the end of danger.

To get the system states space, a formal model is necessary. This model is a formalization of the procedures used during metro line operating, which are textual for the moment. The figure 1 presents all steps of the global approach studied from knowledge acquisition to controllers synthesis, every steps are developed in following sections.

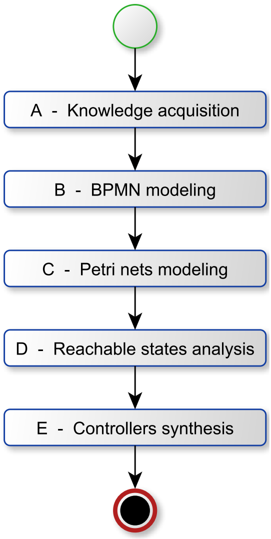
A - Knowledge acquisition
B - BPMN modeling
C - Petri nets modeling
D - Reachable states analysis
E - Controllers synthesis

Fig. 1. Framework

A. Knowledge acquisition

At first, it is necessary to acquire and analyze the dispatchers knowledge and know-how on incidents management procedures.

B. BPMN modeling

To analyze incidents management procedures and their linked evolution, a procedures modeling has to be realized and their interactions to each others have to be studied. BPMN language (Business Process Model and Notation) [10] has been chosen because it gives a graphical representation of these procedures, easily understandable and accessible by dispatchers. It describes business process, among others, exchanges between participants, work flows and sequence of parallel flows.

To study interactions between procedures, resources are identified. These resources are elements describing the metro line operating, like the train position on the line and the presence or not of passengers aboard. Their identification shows the links and constraints between procedures. These resources bring data on the line condition and its evolution in real time. During the incident management, the state of resources evolves with the procedures implementation. With the procedures actions, the dispatcher modifies resources so that the system configuration protects passengers for the best.

C. Petri nets modeling

To analyze the system described before [5], procedures, resources and their interactions are formalized. Petri nets allow to model the procedures parallel evolutions and the links and synchronizations between procedures and resources.

Petri net uses places, transitions and arcs to formalize discrete event systems. A place summarizes all the available information to determine the system behavior. Transitions between states are made instantaneously and events are associated with the firing of transitions.

Controllability informs of possibility to disabled or not the change between two successive configurations of a discrete event system. In traditional Petri net modeling, all transitions are assumed to be controllable and may be prevented from firing by a controller. This information on controllability is given by the events and so this information is known and indicated by the transition.

A transition between two steps of procedures models is controllable because its firing is decided by a human being and may be disabled. For the same reasons, the transition modeling resources state change is also controllable. However, nothing may prevents an incident from occurring, therefore transitions between absence and presence of danger are uncontrollable.

From Petri nets, a system state space may be calculated to set up the reachable states relevant to procedures and resources evolutions. Thus, a reachability graph may be drawn. In this graph, nodes represent system configurations, and each arc represents a transition firing, which transforms one configuration to another and carries the information on controllability.

D. Reachable states analysis

After having defined the danger thanks to a system valuation, the reachability graph allows to identify the states resources combinations which do not ensure the passengers safety. Then, the succession of configurations, the sequence of events, leading to unsafe states are studied in order to try to avoid these states and so to protect passengers. This study is based on Supervisory control theory (SCT) developed by Ramadge and Wonham. Uncontrollable sequences of events not to follow to avoid unsafe states are identified with an algorithm developed in a following section. These sequences of events inform dispatchers on possible consequences of reachable situations and direct towards a better simultaneous incidents management.

The unsafe states identification and the way to avoid them are the two parts of global approach presented in this paper.

E. Controllers synthesis

The setting up of the control in Petri nets modeling and BPMN procedures will be developed and presented in a future paper.

IV. States space analysis

A. Presentation

Supervisory Control Theory (SCT) [14] applies formal reasoning on an uncontrolled process model (the plant) and a desired behavior model of the controlled system (the specification). From plant and specification, a safety device, called a supervisor, can be automatically synthesized. The supervisor controls the plant so that it always stays within the limits defined by the specification, by dynamically disallowing the plant to generate events that may otherwise have been generated.

Initially based on finite automata, supervisory control theory can be applied on systems modeled by Petri nets. For some control objectives, the problem is controlling that the system does not achieve a set of forbidden states and only evolves in good states. Controller has to force the plant so that controlled plant remains in a safe set of states. The sequences of events leading to a forbidden state are analyzed and, depending on their controllability, are authorized or forbidden.

In recent years, some methods have been introduced for avoiding forbidden states and for controller synthesis. A state of the art of this approach is developed in the next section.

B. State of the art

Gaudin, and al. [7] define the set of events sequences to avoid reaching a forbidden state. In this paper, the control of structured plant modeled as asynchronous finite states machines and hierarchical finite state machines is enforced in order to solve the state avoidance control problem. Locally solved, a global supervisor ensuring the global property is provided. In his thesis [6], he generalizes his approach to the forbidden states problem for concurrent systems modeled with safe and conservative Petri nets.

Following papers study the controller definition in Petri nets corresponding to a states interdiction in reachability graph.

The method outlined by Giua and al. in [9] uses conditions associated with controllable transitions to solve the forbidden states problem. It defined Generalized Mutual Exclusion Constraints (GMEC) as a condition that limits a weighted sum of tokens contained in a subset of places.

Theory of regions, which was used in [8], generalizes the previous approach considering controllable and uncontrollable transitions. This method generates some constraints to prevent the system from entering the forbidden states. Solving these constraints generates system control places that obtain maximally permissive behavior.

C. Contribution

All these publications concern the system control with forbidden states: for different kind of systems and to define control places. The set of forbidden states is always a given and invariable control parameter, the way to determine it is not studied. But, in this study, the set of forbidden states is not a known parameter. A method is developed to define and calculate this set.

In the studied system, requirements are not described with desired behaviors, like specifications, but with a set of states to forbid. In this paper, an algorithm is developed to calculate the set of states to forbid corresponding to some special system characteristics. SCT is applied in order to determine the set of states to avoid reaching these states. The difference with usual supremal controllable algorithm is the calculation iterative side of the forbidden transitions.

V. Determining of critical and dreaded states

A. Definitions

To distinguish the notion of forbidden states, presented in the previous section, with the set of states with common characteristics calculated in this paper, an other name is defined: the set of critical states.

The control objective developed in this article is to prevent the system from reaching states of the unsafe region from which it is just possible to go away with a transition from the set of uncontrollable events. To determine this set, the set of critical states, system characteristics considered here are: the inclusion in the defined set of dangerous states and the outgoing transitions controllability. Only controllable events may be forbidden that is why the set of dreaded states, from which it is possible to reach a critical state by a uncontrollable sequence of events, is calculated. To avoid critical states, all controllable transitions which lead to a dreaded state should be banned.

However, this interdiction may modify the states characteristics, like the outgoing transitions, and then the set of critical states. To respect the control objective, the critical and dreaded states calculation should be iterative. To introduce this control, it is necessary to define some notations.

B. Notations

Definition 1: Σ_{uc} is a set of uncontrollable events (these events cannot be prevented from happening by controller) and Σ_c the controllable events, which may be disabled by a controller: $\Sigma = \Sigma_c \cup \Sigma_{uc}$. Σ^* defines a sequence of events.

Definition 2: A deterministic finite automaton G can represent a DES [3]. It is a 5-tuple:

$$G = (Q, \Sigma, \delta, q_0, Q_m)$$

where Q consists in a finite set of states, Σ a nonempty finite set of events, $q_0 \in Q$ the initial state, $Q_m \subseteq Q$ a set of final or marked states. δ is a transition map $\delta : Q \times \Sigma \rightarrow Q$. Let $\sigma \in \Sigma$ an event and the automaton be at the state $q \in Q : \sigma$ may occur only if $\delta(q, \sigma)$ is defined. δ is also defined with $s \in \Sigma^*$

Let $G = (Q, \Sigma, \delta, q_0, Q_m)$ the reachability graph of the system achieved with Petri nets model.

Definition 3: A dangerous state is a state where the system presents factors that may lead to a potential accident realization. It appears following a dangerous event, like an incident. A state is considered as a dangerous state if an incident has been occurred and is still in progress and if resources are in a particular combination defined by system valuation.

We note $Q_d \subseteq Q$ the set of dangerous states of the system G.

Definition 4: A critical state is a state among dangerous states Q_d with only uncontrollable outgoing transitions. The set of critical states of the system G is defined by:

$$Q_{cr}(G, Q_d, \Sigma_{uc}) = \{q \in Q_d \mid \forall \delta(q, \sigma) \text{ defined}, \sigma \in \Sigma_{uc}\}$$

Definition 5: A dreaded state is a state from which it is possible to reach at least one critical state by a uncontrollable sequence of events. By definition, all critical states are included in the set of dreaded states. The set Q_{dr} of dreaded states of the system G is defined by:

$$Q_{dr}(G) = Q_{cr} \cup \{q \in Q \backslash Q_{cr} \mid \exists s \in \Sigma_{uc}^* : \delta(q, s) \in Q_{cr}\}$$

To sum up, critical states are dreaded states, some dreaded states are only dangerous states and some just belong to the system states space. The figure 2 below shows inclusions between sets defined before.

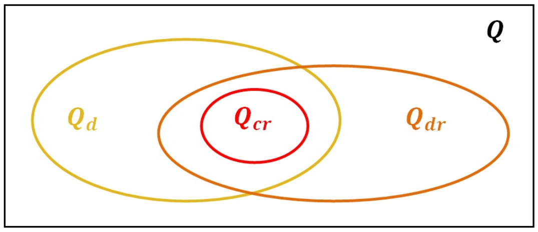

Fig. 2. Sets inclusions

The algorithm presented in the next section aims to determine the sets of dreaded states and so needs to identify their incoming transitions.

Definition 6: The banned transitions correspond to the set of incoming transitions of dreaded states Q_{dr} with a controllable event:

$$\delta_b(G) = \{(q, \sigma, q') \mid q \in Q, \sigma \in \Sigma_c, q' \in Q_{dr} \text{ and } \delta(q, \sigma) = q'\}$$

C. Algorithm

In this paper, the study concerns states included in the set of dangerous states with only uncontrolled outgoing transitions. The control objective leads to ban some controllable transitions and a dangerous state may become a new critical state if its possible remaining outgoing transitions are uncontrollable. The critical and dreaded sets searching should be iterative and carry on until the complete building up of two stable sets.

Indeed, after determining critical and dreaded states during the first iteration, controllable transitions to avoid dreaded states are defined. Banning these transitions in the process, critical set could be extended and therefore there could be more dreaded states. These calculations are iterative until the calculated critical area is complete.

The algorithm below describes all the needed steps to determine critical and dreaded states.

Let $G = (Q, \Sigma, \delta, q_0, Q_m)$ the system reachability graph and Q_d, the subset of all dangerous states of Q.

This algorithm is developed in Python language, based on the tool DESlab [4]. It is a scientific computing program for analysis and synthesis of discrete event systems modeled as automata.

Algorithm 1 Calculate critical states and dreaded states

```
Require: $Q_{cr}(G, Q_d, \Sigma_{uc}) = [\,]$
Require: $G_{test} = G$
Require: $Q_{dr}(G_{test}) = [\,]$
  Calculate $Q_{cr}(G_{test}, Q_d, \Sigma_{uc})$
  while $Q_{cr}(G_{test}, Q_d, \Sigma_{uc}) \neq Q_{cr}(G, Q_d, \Sigma_{uc})$ do
    $Q_{cr}(G, Q_d, \Sigma_{uc}) \leftarrow Q_{cr}(G_{test}, Q_d, \Sigma_{uc})$
    Calculate $Q_{dr}(G_{test})$
    Calculate $\delta_b(G_{test})$
    for all $t \in \delta_b(G_{test})$ do
      Delete $t\ in\ G_{test}$
    end for
    Calculate $Q_{cr}(G_{test}, Q_d, \Sigma_{uc})$
  end while
  return $Q_{cr}(G, Q_d, \Sigma_{uc})$
  return $Q_{dr}(G_{test})$
```

D. Steps of algorithm

To illustrate this algorithm, following theoretical example describes the different steps, the sets calculations and the while loop importance.

The example automaton is a part of a deterministic automaton : only seven states are considered. The transitions not considered here are drawn with dotted lines. For the transitions, controllability is represented by the letters C if controllable and UC if not. Events are not studied nor represented here, only their controllability are examined.

The set of dangerous states is a given system parameter, represented by states colored in yellow. Critical states, circled in red, and dreaded states, with an orange border, are calculated thanks to the algorithm. The transitions leading to a dreaded state by a controllable event are crossed in blue.

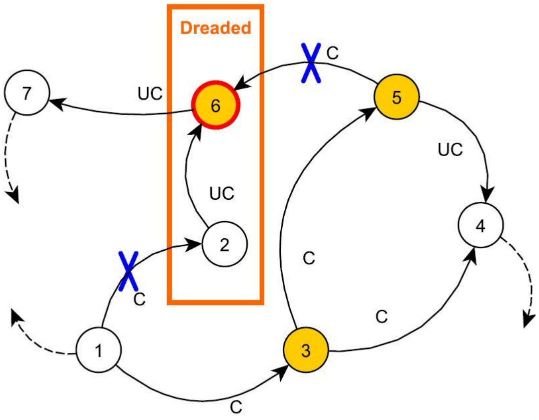

Fig. 3. First iteration

For this example, $Q_d = \{3, 5, 6\}$ defines the set of dangerous states. At the first iteration (figure 3) in the algorithm, $Q_{cr}(G, Q_d, \Sigma_{uc}) = \{6\}$ and $Q_{dr}(G_{test}) = \{2, 6\}$ because the state 2 is the only state which leads to the critical set by an uncontrollable sequence of events. Then $\delta_b(G_{test})$ is calculated and $1 \overset{C}{\to} 2$ and $5 \overset{C}{\to} 6$ have to be banned not to reach the dreaded set. At the end of first iteration, in the controlled system without the banned transitions, the only outgoing transitions of the dangerous state 5 is a uncontrollable event. Moreover, the transition $3 \overset{C}{\to} 5$ is allowed and so it is possible to reach the state 5. Thus, the control objective is not respected.

The calculation of $Q_{cr}(G_{test}, Q_d, \Sigma_{uc})$ shows that an other while loop iteration is necessary.

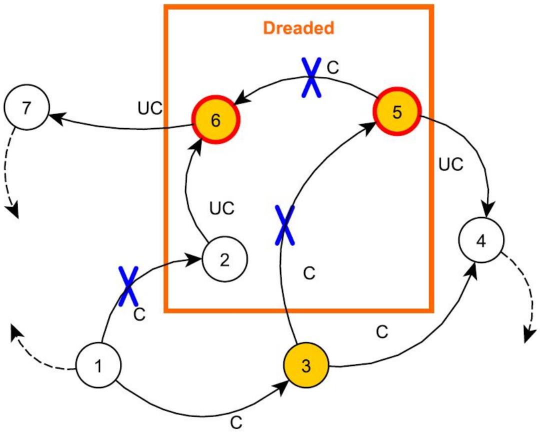

Fig. 4. Second iteration

During second iteration (figure 4), a new critical state appears: $Q_{cr}(G, Q_d, \Sigma_{uc}) = \{6, 5\}$. So the dreaded set is expanded: $Q_{dr}(G_{test}) = \{2, 6, 5\}$. To prevent the system from reaching the dreaded set, a new transition should be banned $3 \overset{C}{\to} 5$, because it allows the system to reach the critical state 5.

At the end of this iteration, dangerous state 3 is still reachable and the outgoing transition $3 \overset{C}{\to} 4$ is controllable.

As conclusion, two iterations are necessary to respect the control objective in this example.

VI. Application: Fire Management and Traction Power Cut

A. Context

The realistic example studies the occurrence of two incidents in a same metro line part: a smoke emission and a person on traffic lanes [11]. Procedures to use for these incidents are the fire management and the traction power cut, to protect the person gone down on traffic lanes.

When a fire is detected on a metro line, the person in charge of the supervision, the dispatcher, has to try to drive trains to a station in order to evacuate passengers and ensure their safety. If a person has been going down on traffic lanes, the metro driver, who witnessed him, has to and will instinctively ask for a power cut. In this studied configuration, with these two incidents, it seems to be possible to have trains with passengers blocked in a smoky tunnel without being able to move to reach a station. This situation is used to apply and validate the algorithm previously presented.

To analyze these two incidents, three resources are identified from the train point of view. These resources are:

- the power supply, on or off, which gives or not the possibility for a train to move,
- the train position on the line, at station or in tunnel,
- the permission to leave a station given by a traffic light turned on or off.

A link exists between the resources power supply and train position. In fact, a train may change its position only if it has a traction power to move. Furthermore, if traffic light is turned on, a train is not allowed to leave the station and change its position.

B. Petri nets modeling (figure 5)

Resource Petri net (columns 3, 4 and 5 in figure 5) is formed by two places, representing the two existing states, and two transitions considered as controllable. Indeed, driver controls the train movement and so its position, dispatcher and drivers may cut the traction power when they want, and traffic lights are turned on or off by the dispatcher.

Like resources, incidents (columns 1 and 7 in figure 5) are modeled by a two places Petri net: presence or absence of danger. Transitions between the two places are uncontrollable, nothing may prevent an incident from occurring or disappearing.

The two Petri nets models of studied procedures are in columns 2 and 6 in figure 5.

In the global system initial state, metro line gets power supply, train is moving in tunnel and traffic lights are turned off. No incident has been occurred yet and the procedures have not begun. This configuration is equivalent to normal operating. System behaviors are represented explicitly by system elements and interactions.

The modeling in Petri nets allows to set up the system state space. The reachability graph shows procedures and resources evolutions and is made up of 288 states and 1360 transitions. The algorithm, defined section V.C, is applied on this finite automaton to find the critical and dreaded sets of states.

C. Algorithm application

To apply the algorithm, three elements are necessary: the reachability graph, transitions controllability and the set of dangerous states. Transitions controllability in the reachability graph is the same as in Petri nets.

A resources combination which protects passengers is identified. When the train is at station and traction power is cut, the two incidents are not very hazardous for passengers. Thus, in this case, danger exists but is low. In all other unsafe states, the resources configuration does not ensure passengers safety. The set of dangerous states Q_d is composed with the unsafe states which are not in a low level of danger.

In this application, a critical state is a dangerous state with only uncontrollable outputs. The only way to leave this kind of states is to wait for the smoke emission end or the person return on a platform.

During algorithm first iteration, only one critical state is identified, the state configuration is:

- smoke emission
- person on traffic lanes
- no power supply
- train in tunnel
- traffic lights turned on
- fire management procedure in progress (step 3)
- traction power cut procedure in progress (step3)

At this iteration, the algorithm detects four dreaded states $|Q_{dr}| = 4$ and, to avoid all these dreaded states, it has to ban sixteen transitions $|\delta_b| = 16$. The table I itemizes the set length for each iterations.

TABLE I
ALGORITHM ITERATIONS RESULTS

Iteration	$\lvert Q_{cr}\rvert$	$\lvert Q_{dr}\rvert$	$\lvert\delta_b\rvert$
n°1	1	4	16
n°2	3	12	28
n°3	5	20	22
n°4	6	24	8

Therefore, four iterations are needed to calculate the complete critical and dreaded sets for this example.

The control objective is to avoid the dangerous states which have only uncontrollable outgoing transitions and may be reached during the incidents management. Finally, to respect this control objective, the algorithm finds six critical states and twenty-four dreaded states.

According to supervisory control theory, a controller, which prevents the system from reaching a critical state, exists. With this controller, no sequence of uncontrollable events may lead to a situation where the passengers safety is not ensured and the risks reduction is not possible.

VII. Conclusions

In this paper, two objectives have been aimed. First, procedures coordination study and resources identification allow to determine the unsafe situations which may be reached during the managements of several incidents. Then, the control objective is to analyze all system reachable situations to react in a controllable way to go to a safer situation. Thus, it is possible to offer solutions to avoid situations where the only way is to wait for the danger end. With this end, the algorithm developed identifies dreaded and critical states reachable during the incidents management to avoid dangerous states and only keep safety sequences of events.

Unlike the existed forbidden states approaches, the algorithm allows to avoid critical states defined by some special system constraints. Here, constraints are the inclusion in a specific set of dangerous states and the outgoing transitions controllability. Algorithm iterative side allows to analyze all system studied configurations and respect the control objective. The computation result is interpreted in terms of management procedures instead of automatic control.

With regard to global approach, this paper is one step to reach the research works objective: analyzing incidents

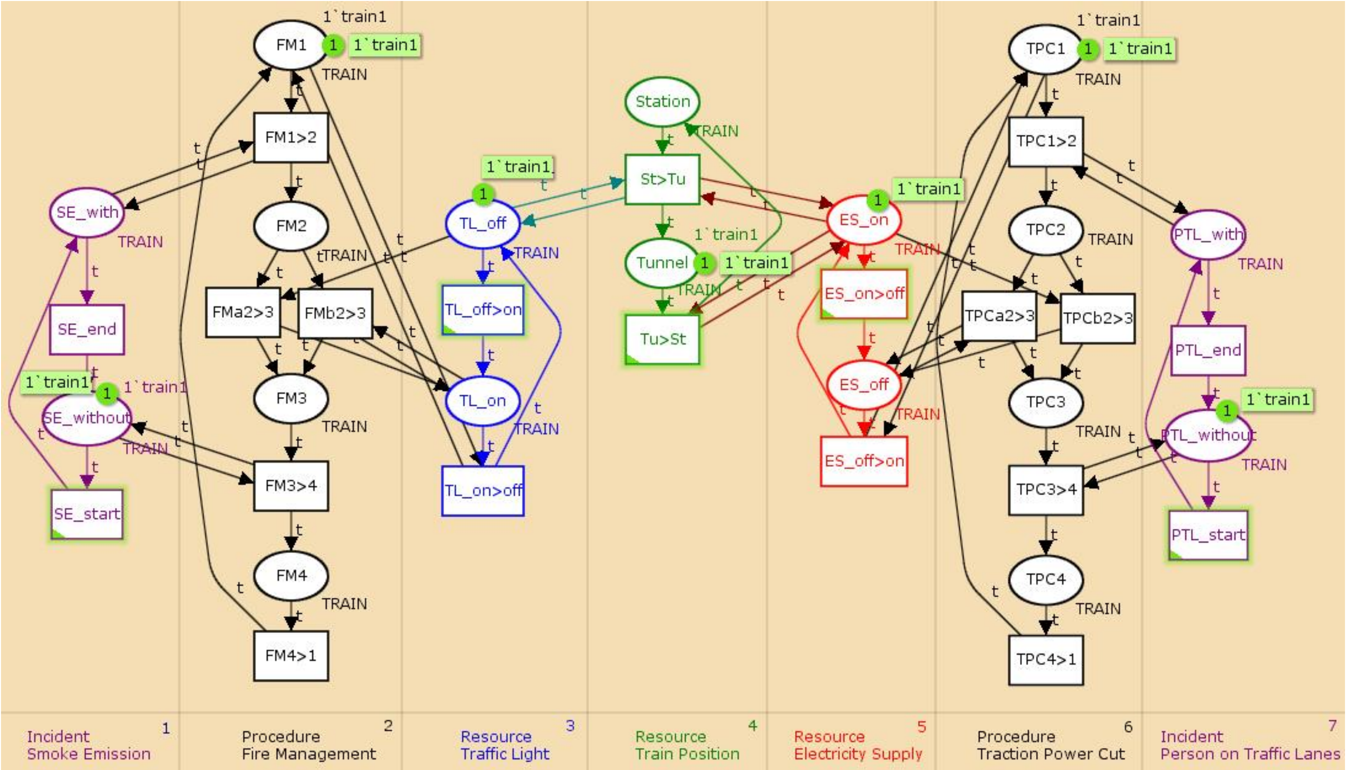

Fig. 5. Petri net model

management procedures with the aim of improving people safety. In fact, this paper goes towards studying procedures coordination, identifying system unsafe configurations and determining sequences of events which could increase the danger for passengers.

To improve the algorithm and get closer to the global objective, the control set is going to be introduced in Petri nets modeling. This control will give a better visualization and understanding of admissible behaviors.

In an off-line incidents management thought in transport systems, a more comprehensive procedures study which considers interactions with each other could enhance passengers safety during a session with incidents. In an on-line application, a decision support system could be developed based on results given by the algorithm. Indeed, the system would inform on the possible reached situation consequences according to the procedures development. It would propose a sequence of events which ensures passengers safety too.

Acknowledgment

The present research work is partially supported by Thales Communications & Security.

References

[1] F. Belmonte, K. Berkani, J. Boulanger, and W. Schon. Safety enhancement of railway traffic by modern supervision systems. In *Seventh World Congress on Railway Research., Montreal (Canada)*, pages 4–8, 2006.

[2] P. Brézillon, C. Gentile, I. Saker, and M. Secron. Sart: A system for supporting operators with contextual knowledge. *Interdisciplinary Conference on Modeling and Using Context (CONTEXT-97)*, 1997.

[3] C.G. Cassandras and S. Lafortune. *Introduction to discrete event systems*, volume 11. Kluwer academic publishers, 1999.

[4] L B Clavijo, JC Basilio, and L Carvalho. Deslab: A scientific computing program for analysis and synthesis of discrete-event systems. In *Discrete Event Systems*, volume 11, pages 349–355, 2012.

[5] R.M. Dijkman, M. Dumas, and C. Ouyang. Formal semantics and analysis of BPMN process models. 2007.

[6] B Gaudin. *Synthèse de contrôleurs sur des systèmes à évènements discrets structurés*. PhD thesis, Rennes 1, 2004.

[7] B. Gaudin and H. Marchand. Modular supervisory control of asynchronous and hierarchical finite state machines. In *In European Control Conference, Ecc 2003*. Citeseer, 2003.

[8] A Ghaffari, N Rezg, and X Xiaolan. Design of a live and maximally permissive petri net controller using the theory of regions. *IEEE transactions on robotics and automation*, 19(1):137–142, 2003.

[9] A Giua, F DiCesare, and M Silva. Generalized mutual exclusion contraints on nets with uncontrollable transitions. In *Systems, Man and Cybernetics, IEEE*, pages 974–979. IEEE, 1992.

[10] Object Management Group. Business process model and notation (BPMN) version 2.0- http://www.bpmn.org/, 2011.

[11] A. Maalel, L. Mejri, and H.H. Mabrouk. Towards an ontology of help to the modeling of accident scenarii" application on railroad transport. *Sciences of Electronics, Technologies of Information and Telecommunications*, 2012.

[12] D Paquereau, L Pietrac, E Niel, and L Bouresche. Démarche de formalisation et de synthèse de procédures d'exploitation d'une ligne de métro. In *Journée des Doctorants MACS*, 2013.

[13] L. Pasquier, P. Brézillon, and J.C. Pomerol. From representation of operational knowledge to practical decision making in operations. *Decision Support throught Knowledge Management*, 2000.

[14] P.J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1):81–98, 1989.