SaNTEA: Stopwatch Petri Net Based Tool for Embedded Systems Analysis

Afifa Ghenai, Hadjer Slimani, and Mohamed Benmohammed

***Abstract*—This paper describes the first version of a reliability analysis tool that implements a new version of feared scenarios generation algorithm. SaNTEA (Stopwatch Petri Net based Tool for Embedded systems Analysis) allows the representation of the suspension and resumption of task execution and the extraction of feared scenarios that lead an embedded system to a critical situation. In this version of SaNTEA, discrete and continuous dynamics of the system are coupled in the same formalism; feared scenarios can be extracted directly from an object-oriented stopwatch Petri net model.**

***Index Terms*—Embedded systems, feared scenarios, object oriented approach, reliability tool, stopwatch petri nets.**

I. Introduction

Today's complex embedded systems are becoming an integral part of our everyday life but their complexity has grown dramatically and more rapidly than the ability to design them and the ability to make them behave as they was designed to. Obviously, more complex systems are less reliable and their critical failures can present significant danger and may have serious consequences including the loss of lives and financial losses. That is why reliability is of vital importance and the major design goal of embedded systems. To ensure this prime requirement, embedded engineers must design an embedded system to run continuously without failures for a long time. Moreover, they must assure that the developed system guarantees timing behavior, especially, the taking into account efficiently of the suspension and resumption of task execution. Thus, to reach a higher level of reliability, system reliability analysis must be performed throughout all stages of its development lifecycle. Also, powerful techniques must be used in the early stages of development in order to prevent or detect and treat failures since the design process, and, novel tools must be developed for automatically assess the reliability status and envisage optimal reconfigurations.

In order to face the increasing complexity of embedded systems, the feared scenarios approach proposed by Khalfaoui [1], improved and implemented by Medjoudj [2] and Sadou [3] allows the determination of the exact conditions of the critical event occurrence and the sequences of actions and state changes leading the system to leave its normal behavior towards the feared state. Critical scenarios which are unknown during the design phase of embedded systems are extracted from a Petri net model without generating the associated reachability graph. In this paper, we present a novel reliability analysis tool allowing the extraction of feared scenarios from a stopwatch Petri net model, a rigorous and powerful tool of reliability analysis with a better expression of temporal behaviors than time Petri nets. Indeed, discrete dynamics of the studied embedded system is modeled using object-oriented stopwatch Petri nets that take into account the interruption and resumption of tasks and allows the identification of more dangerous behaviors of interruptible systems. Then, a hybrid simulation is performed by coupling this discrete dynamics with the continuous dynamics that induces a firing transitions order depending on the dynamic nature.

The rest of this paper is outlined as follows: in Section II, we explain the principle of our feared scenarios generation method and discuss its advantages. In Section III, we describe our reliability analysis tool and present a case study to illustrate the functionalities of SaNTEA. Finally, we conclude the paper in Section IV.

II. Hybrid Feared Scenarios Generation Method

A. Principle

The main idea of the developed reliability tool is to couple the discrete and continuous dynamics in the same stopwatch Petri net model without simulating the entire embedded system. Indeed, hybrid simulation is performed only on the partial order of events leading the studied system to the feared state. In a single calculation, we can see the impact of the physical aspects modeled by a Java code on the discrete behavior modeled by object-oriented stopwatch Petri nets, without exploring the whole reachability graph.

The system modeling in SaNTEA is based on generating feared scenarios from a Post- and Pre-initialized Stopwatch Petri Net (SWPN) [4], an extension of time Petri nets (TPN) by including in its semantics the behavior of interruptible systems. Stopwatch Petri nets have the advantage of combining the concision of Petri nets with the power of stopwatch automata. They include two types of transitions: interruptible transitions and non-interruptible transitions. Pre-initialization is a clock initialization that happens when a non-interruptible is enabled; its associated clock is then initialized. However, post-initialization is a clock initialization depending on transition firing. After being fired, the interruptible transition initializes the clock.

SWPN allow modeling interruptible systems and offer a simple graphic formalism where the modification concerns only the clocks initialization [5]. They are based on more

Manuscript received November 15, 2013; revised January 15, 2014.

The authors are with the Faculty of new technologies of information and communication, University Constantine II, Algeri (e-mail: afifa.ghenai@gmail.com).

DOI: 10.7763/IJCTE.2014.V6.882

simple principles than IHTPN (Time Petri Nets with Inhibitor Hyper arcs) which connect a place to an interruptible transition by an inhibitor arc [6].

The definition of a scenario is based on the concept of event and relations between events [7]. Feared scenarios describe how the system leaves its good functioning and evolves towards a final critical state called: feared state.

Definition 1. (Event): We consider a Petri net (P, T, Pre, Post), M_0 its initial marking. An event is a particular firing of a transition $t \in T$. the set of events is noted E. From M_0, if the transition t_i is fired for the j^{th} time, this is the occurrence of the event e_i^j. [8]

Definition 2. (Scenario): A scenario sc, noted sc = (l, $\prec_{sc}$) associated with the Petri net P and the couple M_0 and M_F markings, is a set of events l provided with a strict partial order $\prec_{sc}$ defined on the events of l. If for $e1, e2 \in l$: $e1 \prec e2$, then the event e1 precedes the event $e2$ in the scenario sc. [8]

B. Hybrid Simulation Steps

The proposed method contains four steps; it is based on combining in the same formalism, a discrete simulation of the studied embedded system without generating the associated reachability graph, by analyzing a more detailed configuration of the system using stopwatch Petri nets and performing a discrete simulation of an algorithm which allows the extraction of more feared scenarios, and, a continuous simulation represented by a Java code. These new scenarios are not taken into account by the preceding feared scenarios approaches. In the first step, nominal states of the studied system must be determined: all places whose marking represents a normal functioning state are either known or obtained by a Monte Carlo simulation. In the second step, target states are determined: a target state is either the feared state or states which have direct or indirect causal relations with the feared state [8]. In the third step, a backward reasoning is performed from the feared state and going up the chain of causalities until we reach a normal functioning state called: the conditioners state.

In the fourth step, a forward reasoning is done starting from the conditioner state in order to determine the exact conditions of the feared behavior occurrence by coupling the discrete dynamics represented by the feared scenarios generation algorithm from a stopwatch Petri net with the continuous dynamics represented by a Java code. Indeed, our reliability tool SaNTEA allows the determination of all the sequences leading to the feared state from the initial conditioner state, and all the bifurcations between the normal functioning and the feared behavior (a bifurcation is a conflict between transitions). The occurrence probability of feared events is calculated using a discrete simulation on the partial orders (represented in red, in Fig. 1.) generated by the feared scenarios algorithm, to avoid the state space explosion.

In order to take into account the suspension and resumption of tasks, SaNTEA proposes a more detailed configuration of the studied system by introducing the stopwatch 'α', the maximum stop time of a task is α_{max}. The resumption of this task cannot be done if its stop time exceeds α_{max}. Indeed, modeling interruptible systems using stopwatch Petri nets allows generating two kinds of bifurcations. The stopwatch value determines the presence of the new kind of bifurcations which are conflicts between transitions of the non-resumption of interruptible transitions and transitions of the resumption (the normal functioning) of interruptible transitions. The firing of interruptible transitions that cannot be resumpted because of non-respect of time constraints leads the system to the feared state and allows generating of new feared scenarios which are not found by the preceding feared scenarios approaches. Fig. 1 shows the proposed hybrid simulation principle.

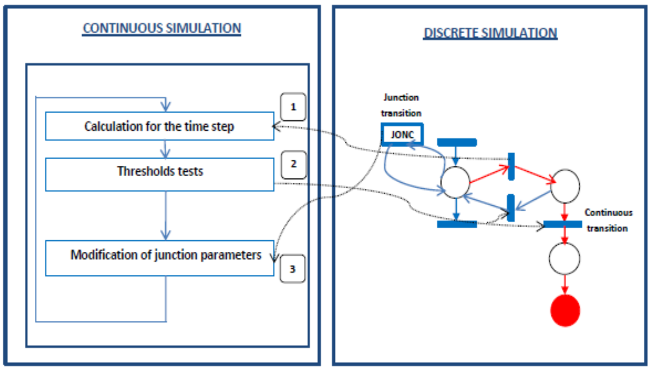


Fig. 1. Hybrid simulation principle.

After initializing all variables of Java code, transitions of continuous dynamics are used when the continuous code resolve the differential equations and execute the procedures of the probability laws. Continuous simulation is performed for the step of time which separates the firing time of the sensitized continuous transition from the firing time of the next transition. This calculation is coupled with the discrete part of our tool SaNTEA, it is represented by (1) in Fig. 1. All thresholds are tested during this step of time, continuous simulation is immediately suspended if a threshold is exceeded and indicates to the discrete part the exceeded threshold time and value, discrete simulation is then continued. If the stop time (α) of an interruptible task which allows the calculation of its suspension duration exceeds the threshold α_{max} which represents the maximum stop time of this task. The continuous calculation indicates the value of α to the discrete calculation, thresholds tests are represented by (2) in Fig. 1. In this case, junction transitions must be fired and the discrete calculation indicates to the continuous part all modifications that must be done in junction parameters, this is represented by (3) in Fig. 1. These parameters are depending on the components which undergo failures, and also, on the transitions representing the non-resumption of an interruptible transition.

III. SANTEA PRESENTATION

In this section, we present the first version of SaNTEA. It is a French version coded in Java language.

A. TINA

We use the version 3.0.6 of the tool TINA (TIme Petri Net Analyzer) [9] developed in LAAS-CNRS laboratory. It allows the analysis, the textual and graphical description of Petri nets. Indeed, Textual files of stopwatch Petri nets generated by the module *TEXTIFY* of TINA [10] are input files to our tool SaNTEA. Also, *struct* files generated by the module *structural analysis* are used because the marking enrichments are not possible if the system consistency is not preserved. This is determined by the marking invariants.

B. User Interface

Fig. 2 shows the user interface of SaNTEA. It is composed of four zones.

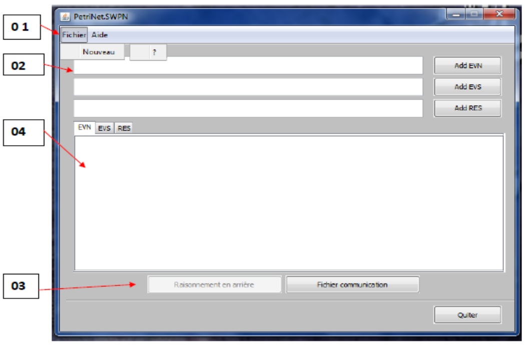


Fig. 2. SaNTEA user interface.

In the first zone, there are two buttons. The first button allows data reinitialization; the second allows the use of SaNTEA manual. The second zone is used to extract and display the model files. In the third zone, there are two buttons. The first button is used to display the results of backward reasoning. The second button is used to build the communication file between objects. The fourth zone is used to display the analyzed files.

C. Case Study

In order to guarantee a best level of vehicle reliability, the mechatronic automobile system presented in [11], is studied in our tool SaNTEA during the design phase of the antilock braking system (ABS), the main safety component in vehicles by preventing wheels lock up. In order to avoid wheels lock up, nose wheels are controlled by an electronic control unit that orders the appropriate modulator valves to decrease brake pressure. Then, the computer actuates the valves of the brake system, according to the received information. If computer logic and sensors identify potential wheel lock up or a difference between the vehicle speed and wheels speed, hydraulic actuators decrease the pressure of the braking liquid to a safe level, until the wheel starts to turn or until there is no more difference in measured speed. Only the brake pressure of wheels that are in danger of locking up is adjusted. Many works related to antilock braking systems design have been proposed [12]-[14].

SaNTEA allows a best expression of temporal behaviors by representing the suspension and resumption of tasks and more interactions between the ABS components. Indeed, this more detailed configuration of the antilock braking system provides more interaction between its objects and more feared scenarios that cannot be generated by the previous feared scenarios approaches and when the old ABS model presented in [11] is used. The studied antilock brake System (ABS) is presented in the Fig. 3 [8].

In the proposed object oriented approach, two classes are used in order to model the functional and dysfunctional behaviors of the antilock braking system using stopwatch Petri nets: The first class is the common block of all the components of the system: the piston, brake pedals, brake fluid, fluid reservoir, the calculator with software, brake discs and brake pads. Fig. 4 presented in [8], shows a stopwatch Petri net representation of the common block model.

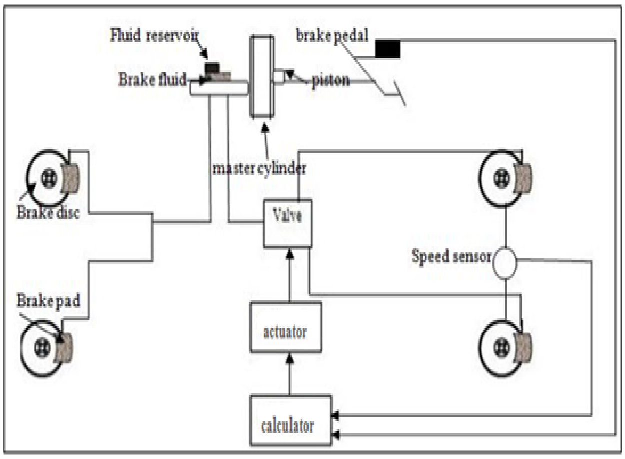


Fig. 3. Antilock braking system (ABS).

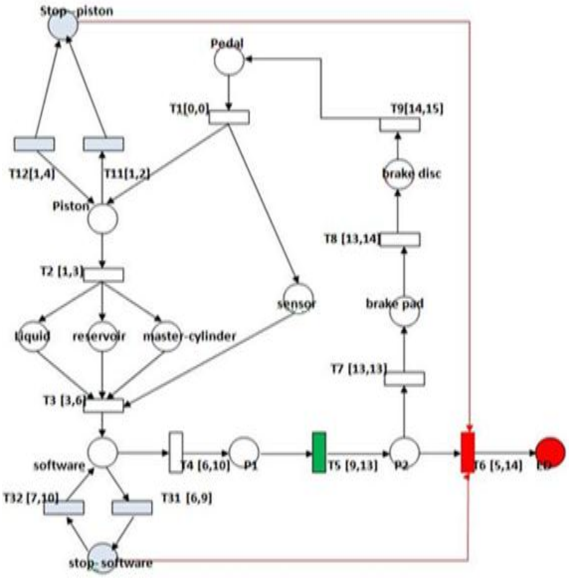


Fig. 4. SWPN model of the common block.

Normal states are represented with places and transitions in white. The suspension and resumption of tasks are represented with places and transitions in blue. The feared states are represented with places and transitions in red. Finally, the call of ABS object is represented with the transition in green. The suspension of tasks in the common block can occur either in the piston or the software. In the piston, the firing of the transition T11 in the time interval [1], [2] represents a suspension of task. The resumption of this task is represented by the firing of the transition T12 in the time interval [1], [4]. However, if the failure duration exceeds this time interval, the system leaves its normal functioning towards a feared state.

In the software, the firing of the transition T31 in the time interval [6], [9] represents a suspension of task. The resumption of this task is represented by the firing of the transition T32 in the time interval [7], [10]. However, if the failure duration exceeds this time interval, the system leaves its normal functioning towards a feared state.

In the continuous part of the common block, the failure probability law of the piston is a Weibull law. The software has an exponential probability failure law. Failures laws allow the determination of junction functions. Table I shows the piston and software failures, respectively.

TABLE I: PISTON AND SOFTWARE FAILURES

Object	Probability law	Failure rate
Piston	Weibull	1000
Software	Exponential	λ_1=5,04.10^{-4} h^{-1}

The second class is the optional block that contains the valve and the actuator. When the valve is open, the liquid increases in the brake pedals and can cause wheels lock up. Consequently, the computer orders the actuator to prevent the pressure from increasing in the circuit. Fig. 5 shows the stopwatch Petri net model of the optional block.

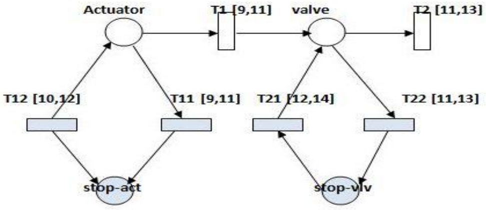


Fig. 5. SWPN model of the optional block.

In the actuator, the firing of the transition T11 in the time interval [9], [11] represents a suspension of task. The resumption of this task is represented by the firing of the transition T12 in the time interval [10], [12]. However, if the failure duration exceeds this time interval, the system leaves its normal functioning towards a feared state. In the valve, the firing of the transition T22 in the time interval [11], [13] represents a suspension of task. The resumption of this task is represented by the firing of the transition T21 in the time interval [12], [14]. However, if the failure duration exceeds this time interval, the system leaves its normal functioning towards a feared state. When the common object (the common block) calls the ABS object (the optional block), the actuator sends a request to the valve to close the brake system. However, if a component failure occurs, the circuit remains open and the system leaves its normal functioning towards the feared state: wheel lock up.

In the continuous part of the optional block, the failure probability law of the actuator is an exponential law. The valve has a Weibull probability failure law. Table II shows the actuator and the valve failures, respectively.

TABLE II: ACTUATOR AND VALVE FAILURES.

Object	Probability law	Failure rate
Actuator	Exponential	λ_1=7,5.10^{-3} h^{-1}
Valve	Weibull	100

1) *Input files*

The graphical and textual descriptions of the different objects (instances of the two classes) of the antilock braking system are input files to our tool SaNTEA. Fig. 6 and Fig. 7 show a graphical description and a textual description of the common block using stopwatch Petri nets, respectively.

Fig. 8 and Fig. 9 show a graphical description and a textual description of the optional block using stopwatch Petri nets, respectively.

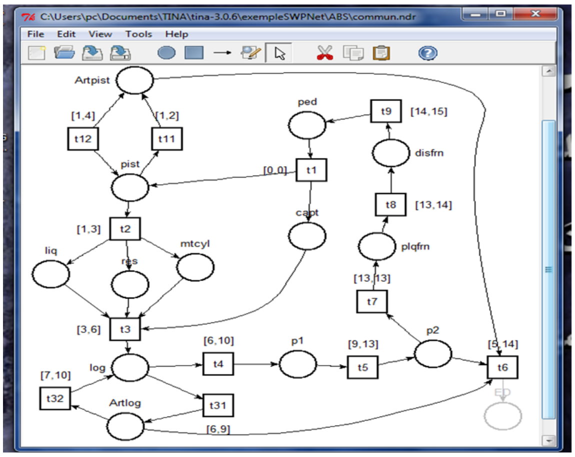
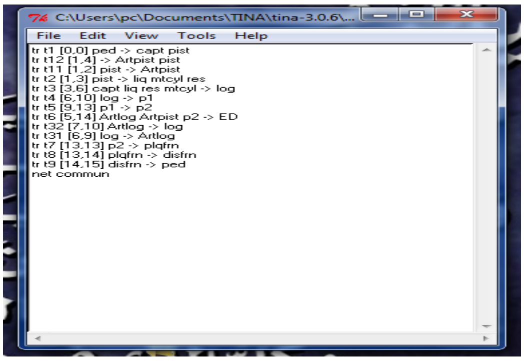


Fig. 6. Graphical description of the common block.


C:\Users\pc\Documents\TINA\tina-3.0.6\...
File Edit View Tools Help

```
tr t1 [0,0] ped -> capt pist
tr t12 [1,4] -> Artpist pist
tr t11 [1,2] pist -> Artpist
tr t2 [1,3] pist -> liq mtcyl res
tr t3 [3,6] capt liq res mtcyl -> log
tr t4 [6,10] log -> p1
tr t5 [9,13] p1 -> p2
tr t6 [5,14] Artlog Artpist p2 -> ED
tr t32 [7,10] Artlog -> log
tr t31 [6,9] log -> Artlog
tr t7 [13,13] p2 -> plqfrn
tr t8 [13,14] plqfrn -> disfrn
tr t9 [14,15] disfrn -> ped
net commun
```

Fig. 7. Textual description of the common block.

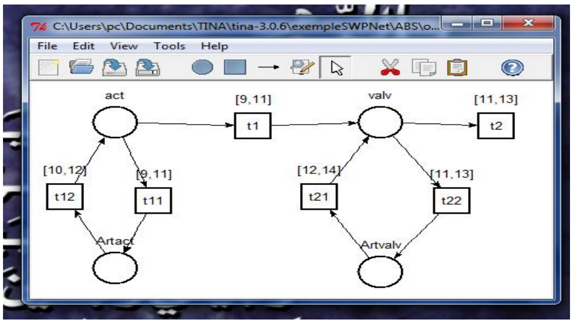
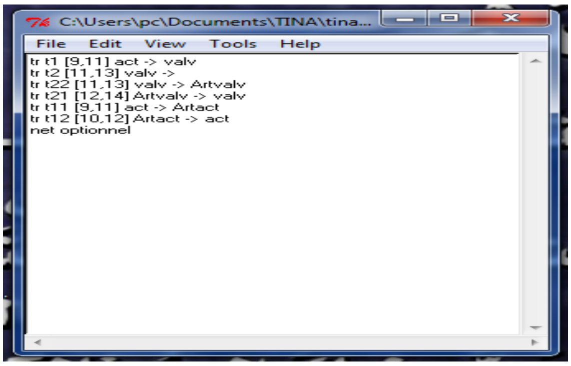

C:\Users\pc\Documents\TINA\tina-3.0.6\exempleSWPNet\ABS\o...
File Edit View Tools Help
act
[9,11]
t1
valv
[11,13]
t2
[10,12]
t12
[9,11]
t11
[12,14]
t21
[11,13]
t22
Artact
Artvalv

Fig. 8. Graphical description of the optional block.

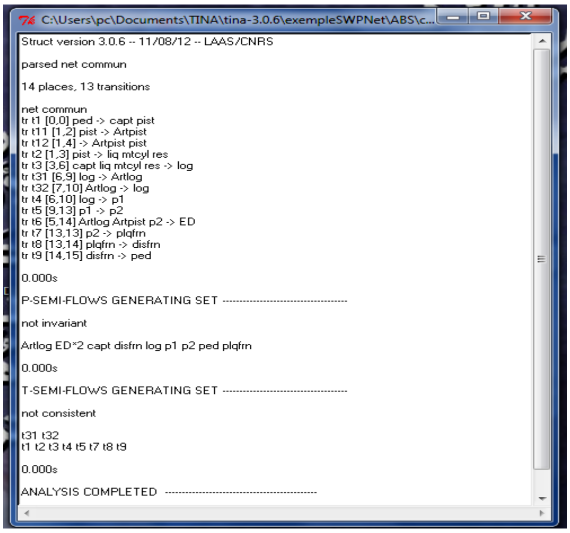


```
tr t1 [9,11] act -> valv
tr t2 [11,13] valv ->
tr t22 [11,13] valv -> Artvalv
tr t21 [12,14] Artvalv -> valv
tr t11 [9,11] act -> Artact
tr t12 [10,12] Artact -> act
net optionnel
```

Fig. 9. Textual description of the optional block.

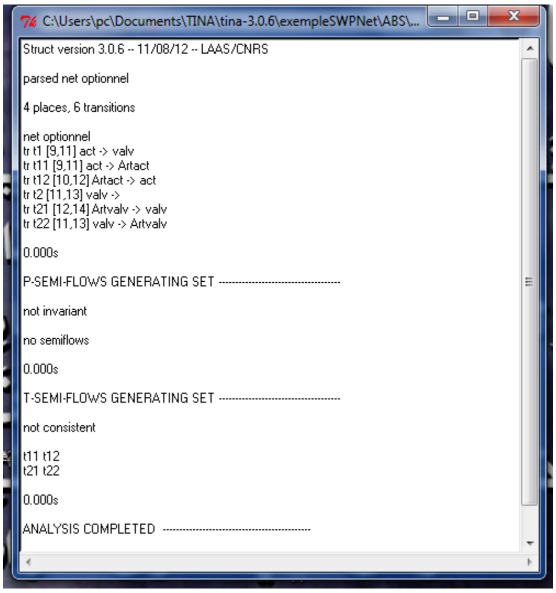
2) *Structural analysis*

The marking invariants are offered by the structural analysis of TINA tool (TIme Petri Net Analyzer). These invariants determine if the system consistency is preserved or not. The marking invariants are necessary before carring out the marking enrichments during the feared scenarios generation algorithm. Fig. 10 and Fig. 11 presented below shows the structural analysis files of the common block and the optional block, respectively.


C:\Users\pc\Documents\TINA\tina-3.0.6\exempleSWPNet\ABS\c...

```
Struct version 3.0.6 -- 11/08/12 -- LAAS/CNRS

parsed net commun

14 places, 13 transitions

net commun
tr t1 [0,0] ped -> capt pist
tr t11 [1,2] pist -> Artpist
tr t12 [1,4] -> Artpist pist
tr t2 [1,3] pist -> liq mtcyl res
tr t3 [3,6] capt liq mtcyl res -> log
tr t31 [6,9] log -> Artlog
tr t32 [7,10] Artlog -> log
tr t4 [6,10] log -> p1
tr t5 [9,13] p1 -> p2
tr t6 [5,14] Artlog Artpist p2 -> ED
tr t7 [13,13] p2 -> plqfrn
tr t8 [13,14] plqfrn -> disfrn
tr t9 [14,15] disfrn -> ped

0.000s

P-SEMI-FLOWS GENERATING SET -----------------------------------

not invariant

Artlog ED*2 capt disfrn log p1 p2 ped plqfrn

0.000s

T-SEMI-FLOWS GENERATING SET -----------------------------------

not consistent

t31 t32
t1 t2 t3 t4 t5 t7 t8 t9

0.000s

ANALYSIS COMPLETED ------------------------------------------
```

Fig. 10. Structural analysis of the common block.



```
Struct version 3.0.6 -- 11/08/12 -- LAAS/CNRS

parsed net optionnel

4 places, 6 transitions

net optionnel
tr t1 [9,11] act -> valv
tr t11 [9,11] act -> Artact
tr t12 [10,12] Artact -> act
tr t2 [11,13] valv ->
tr t21 [12,14] Artvalv -> valv
tr t22 [11,13] valv -> Artvalv

0.000s

P-SEMI-FLOWS GENERATING SET -----------------------------------

not invariant

no semiflows

0.000s

T-SEMI-FLOWS GENERATING SET -----------------------------------

not consistent

t11 t12
t21 t22

0.000s

ANALYSIS COMPLETED ------------------------------------
```

Fig. 11. Structural analysis of the optional block.

D. *Results*

The application of the proposed hybrid simulation to the antilock braking system provides more interactions between the common block and the optional block of the ABS modeled using object-oriented stopwatch Petri nets. The best taking into account of time constraints by representing the suspension and resumption of tasks identifies a new kind of bifurcations between the interruptible transitions and allows the extraction of new feared scenarios leading to the feared state: wheels lock up. These scenarios cannot be generated by the previous approaches when the system is modeled by time Petri nets.

In the first step, discrete simulation is performed in our tool SaNTEA, begins in order the determination the ABS nominal states: places in white (in Figure) and whose marking represents a normal functioning state. Discrete simulation allows the calculation of the feared events occurrence probability in the partial orders generated by the feared scenarios algorithm. All variables of the continuous simulation are initialized in order to draw a random of Monte Carlo simulation.

In the second step, the target and feared state: wheels lock up, is identified.

In the third step, the backward reasoning is made in our tool SaNTEA, from the target state: wheels locking until normal functioning states (conditioner states) are reached: stop-piston and stop-software.

In the fourth step, the forward reasoning starts from the two conditioner states: stop-piston, stop-software (or: stop-vlv, stop-act). Consequently, these input places of the transition T6, are marked. This marking is the cause of the second kind of bifurcation between the interruptible transitions. Indeed, if the place stop-piston is marked (this marking causes a conflict between the transitions T12 and T6). A task is then interrupted: the piston task. This suspension is memorized during the front reasoning. Thus, the continuous simulation is performed for a step of time because the associated continuous transition is sensitized. Differential equations of the piston are resolved and the procedure of the Weibull law is executed.

If the duration of the piston task suspension exceeds the specified time interval, the resumption transition T12 cannot be fired and the stopwatch value is memorized. Indeed, α_{max} is a threshold representing the maximum stop time of a task. If α exceeds α_{max}, the continuous simulation suspends its process and indicates to the discrete part this exceeded value and time of the over threshold, the discrete simulation will continue its process. The associated junction transition is fired and the discrete simulation indicates to the continuous simulation all modification of the junction parameters that must be done.

Consequently, the ABS leaves its normal functioning and the place ED (the feared state: wheels lock up) is marked by the firing of the transition T6. In this case, time constraints must be modified to avoid the drift towards the feared state: wheels lock up. Four scenarios leading to wheels lock up are generated by SaNTEA. Fig. 12 and Fig. 13 show the first and the second feared scenarios, respectively [8].

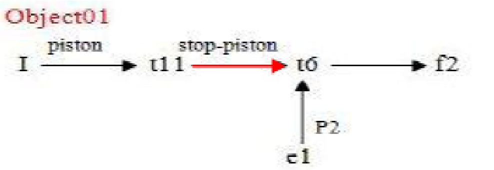


Fig. 12. The first feared scenario.

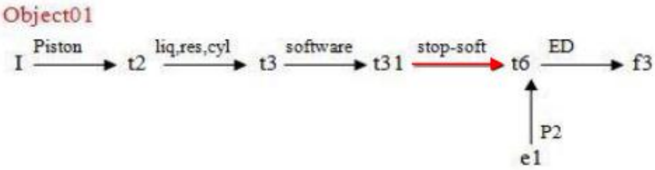


Fig. 13. The second feared scenario.

The new interactions between the common block and the optional block allow the generation of two other sequences of actions that lead the ABS to a critical situation. Fig. 14 and Fig. 15 show these feared scenarios [8].

The third feared scenario is due to the interaction between the suspension of the piston, the suspension of the software, and, the suspension of the actuator.

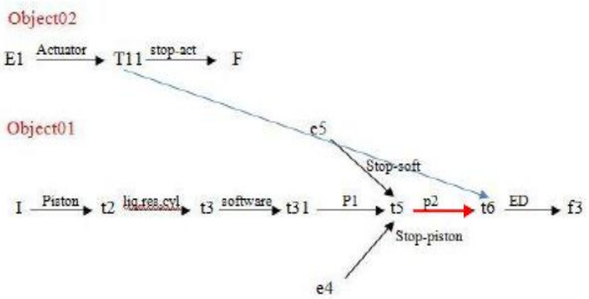


Fig. 14. The third feared scenario.

The fourth feared scenario is due to the interaction between the suspension of the piston, the suspension of the software, and, the suspension of the valve.

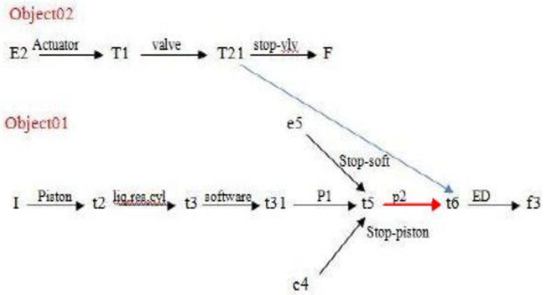


Fig. 15. The fourth feared scenario.

340 histories have been performed in SaNTEA using the proposed hybrid simulation. Table III shows the obtained results of the feared state: wheels lock up.

TABLE III: HYBRID SIMULATION RESULTS

Time (minutes)	Medium	Standard deviation
100	0.144	0.288
200	0.238	0.478
300	0.306	0.519

Fig. 16 shows the occurrence frequency of feared events during 4000 minutes.

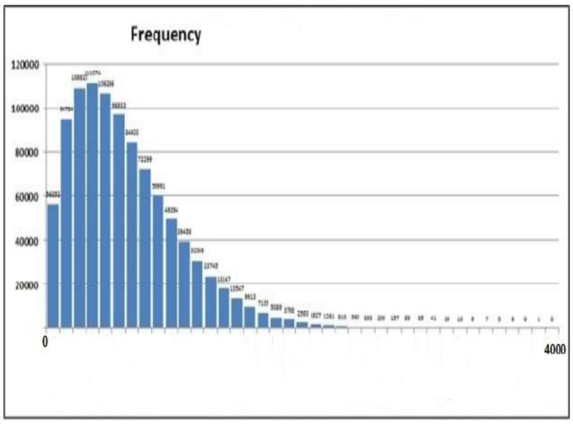


Fig. 16. Occurrence frequency of feared events.

Fig. 17 shows the probability density of feared events obtained by a hybrid simulation performed during 4000 minutes.

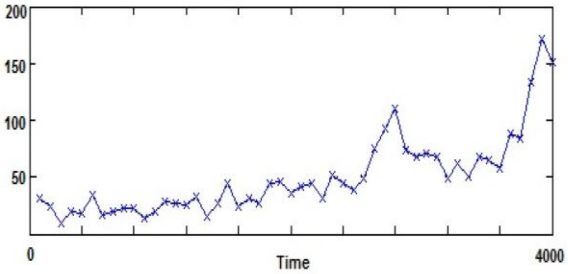


Fig. 17. Probability density of feared events.

IV. CONCLUSION

In this paper, we presented the first version of SaNTEA tool that implements a hybrid feared scenarios generation algorithm. The advantage of its system modeling is the taking into account of the suspension and resumption of tasks using stopwatch Petri nets and the coupling in the same formalism of discrete and continuous simulations. In order to illustrate the functionalities of SaNTEA, the proposed hybrid simulation is applied during the design phase of an antilock braking system (ABS). Our tool provides a more detailed configuration of the ABS using object-oriented stopwatch Petri nets for a best taking into account of time constraints. Thanks to the best expression of temporal behaviors and hybrid simulation, more interactions between the ABS objects are given and more feared scenarios are generated because of non-respect of time constraints. These new scenarios cannot be extracted by the previous feared scenarios approaches using the old ABS model.

REFERENCES

[1] S. Khalfaoui, "Méthode de recherche des scénarios redoutés pour l'évaluation de la sûreté de fonctionnement des systèmes mécatroniques du monde automobile," Thesis, Institut National Polytechnique, Toulouse, France, 2003.

[2] M. Medjoudj, ''Contribution à l'analyse des systèmes pilotés par ordinateurs: Extraction de scénarios redoutés et vérification de contraintes temporelles," Thesis, Paul Sabatier University, Toulouse, France, 2006.

[3] N. Sadou, "Aide à la conception des systèmes embarqués sûrs de fonctionnement," Thesis, Toulouse III University-Paul Sabatier, France, 2007.

[4] A. Allahham and H. Alla, "Réseaux de Petri à chronomètres Post et Pré initialisés, in $6^{ème}$ colloque francophone sur la modélisation des systèmes réactifs," Lyon, France, 2007.

[5] M. Magnin, P. Molinaro, and O. H. Roux, "Expressiveness of petri nets with stopwatches. Dense-time part," *Fundamenta Informaticae*, 2009.

[6] O. H. Roux and D. Lime, "Time petri nets with inhibitor hyperarcs. formal semantics and state space computation," in *Proc. International Conference on Applications and Theory of Petri Nets*, Italy, 2004.

[7] A. Ghenai, M. Y. Badaoui, and M. Benmohammed, "Reliability assessment of embedded systems using stopwatch petri nets," *International Journal of Computer Science, Engineering and Applications*, vol. 2, no. 5, pp. 33-48, 2012.

[8] A. Ghenai, M. Y. Badaoui, and M. Benmohammed, "Towards a good ABS design for more reliable vehicles on the roads," *International Journal of Computer Science & Information Technology*, vol. 5, no. 3, pp. 129-142, 2013.

[9] TINA. [Online]. Available: http://www.laas.fr/tina/

[10] B. Berthomieu and F. Vernadat, "Time petri nets analysis with TINA," in *Proc. 3rd Int. Conf. on The Quantitative Evaluation of Systems (QEST 2006)*, IEEE Computer Society, 2006.

[11] A. G. Mihalache, "Modélisation et évaluation de la fiabilité des systèmes mécatroniques: application sur système embarqué," Thesis, Angers University, France, 2007.

[12] A. Soliman and M. Kaldas, "An Investigation of anti-lock braking system for automobiles," SAE Technical Paper, doi: 10.4271/2012-01-0209, 2012.

[13] Y. Jing, Y. Mao, G. M. Dimirovski, Y. Zheng, and S. Zhang, "Adaptive global sliding mode control strategy for the vehicle antilock braking systems," in *Proc. the 28th American Control Conference*, St Louis, Missouri, 2009.

[14] H. Zhang, F. Fashan Yu, and X. Wang, "Condition monitoring of rope-less elevator braking system based on wavelet denoising," *Journal of Computers*, vol. 8, no. 3, Academy Publisher, doi: 10.4304/jcp.8.3.741-748, 2013.

Afifa Ghenai is a teacher at Faculty of New Technologies of Information and Communication, University Constantine II, Algeria, and, he is a member of LIRE laboratory. Her research domains are embedded systems, formal methods and hybrid simulation.

Hadjer Slimani has a master degree in computer science from University Constantine II, Algeria in 2013. Her research domains are real time systems and formal methods.

Mohamed Benmohammed is a researcher and a professor at the Department of Software Technologies and Information Systems, Faculty of New Technologies of Information and Communication, University Constantine II, Algeria. He received his master degree in computer science in 1988 from HCR, Algiers, and, his PhD degree in 1997 from the University of SBA, in cooperation with the TIMA/CNRS-INPG laboratory, Grenoble, France. He is the head of the AS group of LIRE laboratory. He has published many articles in International Journals and Conferences and supervised many Master and PhD students. His research interests include embedded systems, simulation and formal methods.