Stochastic activity networks for the performance evaluation of fault tolerant systems

Samia Maza

Stochastic activity networks for the performance evaluation of fault tolerant systems

Samia MAZA
Université de Lorraine
Centre de Recherche en Automatique de Nancy
2 avenue de la Forêt de Haye, 54500 Vandoeuvre lès Nancy
samia.maza@univ-lorraine.fr

***Abstract*—The need for high performance, productive and dependable industrial systems has conducted to the design of complex operational architectures. Many features and subsystems are employed and added to a system to produce a fault-tolerant system. All of these subsystems will interact with each other to improve the global system's performances. Each feature/component has its own dynamic and performances and the interactions between them may induce a difficulty to control and evaluate the system's behavior. This paper proposes the use of an integrated modeling approach for fault tolerant systems based on stochastic activity networks. This approach allows the evaluation of some system's performance parameters like the reliability, availability and maintenance costs.**

***Index Terms*—Reliability, Diagnosis, Reconfiguration, Maintenance, Stochastic activity networks, Monte Carlo Simulation.**

I. INTRODUCTION

There are two major complementary approaches to design performances and dependable industrial systems:

- *Off-line approaches*: during the design stage, the dependability factors *RAMS* (reliability, availability, maintainability and safety) are computed using various analysis tools and methods. This allows the implementation of material architectures that meet the RAMS requirements.
- *On-line approaches*: some sub-systems that insure key functions like monitoring, fault detection and isolation, system reconfiguration and maintenance, are used to enhance the system's behavior when faults/failures occur.

Usually, these two studies are not conducted jointly while it is clear that each one is tightly connected to the other and influenced by it. Indeed, as explained before many-subsystems co-exist in fault tolerant (*FT*) systems, like the supervision or diagnosis system and backup systems. The role of the supervision system is to diagnose the occurrence of faults, *i.e.*, to detect and localize the system's faults. Fault diagnosis in fault tolerant systems allows some recovering actions like reconfiguration and maintenance. Material reconfiguration is employed if backup components/systems are used. In the case of fault tolerant control, the diagnosis procedure allows the system to switch from one control law to another according to the system's architecture obtained after the fault occurrence [1]. Maintenance actions can also be undertaken the repair the components which faults have been diagnosed. All these functions and subsystems interact with each other to improve the dependability of the resulting fault tolerant system. Nevertheless, they are not totally reliable and therefore, their performances should be taken into account when assessing the dependability of the FT-system.

Consider the diagnosis procedure for example; it is based on some tuning parameters (*eg*. The threshold) which has an important impact on the quality of the detection. Thus, it also has an impact on the actions needed to recover from faults/failures like reconfiguration and maintenance. Consequently, the diagnosis performances should be considered explicitly when making the dependability study [2][3].

In the same way, the dependability information and objectives could be considered in fault detection and isolation (FDI) procedures, to improve the decision making as well as system reconfiguration [1].

Thus, it appears that an integrated modeling and analysis approach is more suitable to deal with fault tolerant systems [4]. This implies the need of a powerful modeling formalism that can cover in one hand the modeling of deterministic dynamical systems/functions, like the control and diagnosis as well as dynamic behaviors like reconfiguration. On the other hand, this formalism must allow the modeling of probabilistic phenomena like fault/failure occurrence to make a dependability analysis. Notice that actions like reconfiguration and maintenance will make changes the system's structure and thus, in its reliability evaluation model which sets the problem of dynamic reliability assessment. Indeed, the concept of dynamic reliability aims to take into account the interactions between the dynamic and functional behaviors (deterministic behavior) of a system, with its dysfunctional one (stochastic behavior) [5]. It covers the study of systems for which the reliability model evolutes with time under the effect of some random events (a cold redundancy component which switches-on when the principle component fails for example) or when some physical continuous variables cross some thresholds which may induces some functioning mode's

changes [6]. Fault-tolerant systems have such complex behavior and are subject to changes mentioned before. The traditional tools of the reliability can not be applied effectively to solve these problems because they assume an invariant structure in time for the system. Because of complex behaviors of FT-systems, analytical models are generally not able to model all these behaviors and the interactions between them [6].

In, [3] we proposed an integrated modeling approach based on stochastic activity networks, which allows the systematic construction of SAN-models of fault tolerant systems. This approach allows the modeling of dynamic and stochastic behaviors while including explicitly the diagnosis performances. Such modeling allowed us to study in simulation, the impact of the diagnosis parameters and performances on the mean availability.

In this paper, this modeling approach is extended and completed to cover other performances amounts computation like the system's reliability and maintenance costs. The paper is organized as follows:

Section 2 presents the stochastic activity networks and their features. Section 3 is devoted to the brief description of fault tolerant systems. The SANs based modeling approach is explained on an example of an automated thermal process in section 4. The results of Monte Carlo simulations of this process are presented in section 5, where some performances amounts are studied for various diagnosis parameters, maintenance and reconfiguration policies. Section 6 finally concludes the paper.

II. The Stochastic Activity Networks Formalism

Stochastic activity networks (SANs) are discrete events systems modeling formalism, like *Petri* nets and automata. They are also able to model stochastic phenomena and are very similar to the generalized stochastic *Petri* nets (GSPNs). Since this latter is more known and for sake of simplicity, formal definition of GSPNs will be given first. Additional SANs features will be then explained in comparison to GSPNs.

A stochastic Petri net (SPN) is a directed bipartite graph, defined by the six-tuple $SPN = (P, T, I, O, M_0, \Lambda)$. Here T and P are two distinct sets of vertexes. $T=\{T_1,\ldots,T_n\}$ is a set of transitions and $P=\{P_1, P_2, \ldots P_m\}$ is a set of places. A transition can be seen as an event or an action, and a place represents either a condition for the event or a consequence of it. $I \subset PxT$ is the set input arcs and $O \subset TxP$ is the set of output arcs. M_0 is the initial marking vector, $M_0 = (M_0(p_1), M_0(p_2), \ldots M_0(p_m))^T$, where $M_0(P_i)$ is the initial number of tokens of place P_i. $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ is a vector of (a possibly marking dependant) firing rates associated with transitions [7].

A *GSPN* is an *SPN* in which some transitions are timed, while others are immediate. Random, exponentially distributed firing delays are associated with timed transitions, whereas the firing of immediate transitions takes place in zero time, with priority over timed transitions. The selection among possibly conflicting enabled immediate transitions is made through firing probabilities forming the so-called *random switches* [8].

The SANs were first introduced by Mogavar and Meyer in, [9] to model a wide range of systems to make their analysis and performance and dependability assessment [10]. Compared to Petri nets, they are characterized by the following elements:

- *Places:* as for Petri nets, the set of places with their markings can be seen as state of the modeled system. In SANs formalism, places are of two types: ordinary and extended. A marking of an extended place is not the number of tokens like for ordinary places, but can be a variable of any type: array, matrix or a data structure. They can be seen like colored places in colored Petri nets [11], but are different since for extended places, tokens are materialized as variables associated with these places and thus, can not be removed or added to the extended places. Only their value can be red or changed. In *Mobïus* software tool which supports the SAN formalism, blue circles present ordinary places while orange circles present extended places (Fig. 1).
- *Activities:* are equivalent to transitions in *Petri* nets. Unlike GSPNs, in SANs the timed activities can have either a deterministic or a stochastic duration. Stochastic ones are not necessarily exponentially distributed. An activity may complete (i.e., fire) through many possibilities modeled by the so called cases probabilities.
- *Cases probabilities:* they model the uncertainty about the active (or enabled) activity to complete. In GSPNs, it is equivalent to have many enabled transitions in conflict. The conflict being solved thanks to their associated probabilities. In SANs, both timed and immediate activities can have cases probabilities. These latter are graphically represented by small circles on the right side of an activity (Fig. 1). These probabilities can be marking dependant and their sum should be equal to one.
- *Input gates:* used to control the activation of activities. An input gate defines the condition on the marking of its input place to make the activity enabled. It also defines, thanks to the *input function*, the new marking of these places after the completion of its associated activity. When connected to an extended place, it allows the reading of its associated data (marking). An input gate is modeled by a red triangle.
- *Output gates*: they define, thanks to their *output function*, the marking change on the output places of an activity when it completes. When connected to an extended place, it allows the writing over its associated data.

Here after is given an example of a SAN model with all its described elements.

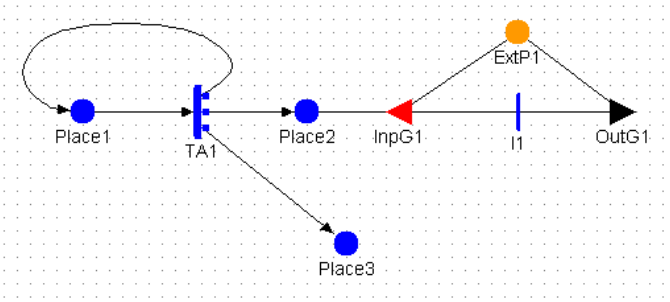

Fig. 1. An example of a SAN model with all its features.

In (Fig. 1) the timed activity *TA1* has three cases probabilities for three possible completions of *TA1*. the instantaneous activity *I1* has an input and output gate (resp. *InpG1* and *OutG1*). The input gate has two input places: an ordinary place (*Place2*) and an extended one (*ExtP1*). This latter couldn't be connected directly to an activity, but only to its input and output gates. If *ExtP1* is associated to a float variable, *x*, then this latter can be written as $x=M(ExtP1)$, where $M(ExtP1)$ is the marking of place *ExtP1*. The output gate *OutG1* allows to change the value of the variable *x* (for example $x=x^2+1$) thanks to the *output function*.
Extended places will be used later to evaluate some performances amounts like maintenance costs.

III. Fault Tolerant Systems

Fault-tolerance is the property that enables a system to continue operating properly in the event of failure/fault of its components. Fault-tolerance is particularly sought-after in high-availability or life-critical systems.

To achieve fault-tolerant systems, many subsystems and functionalities are added to the nominal system to reduce the perturbations' impact on the system's behavior and performances. The main FT-systems functionalities are briefly explained hereafter.

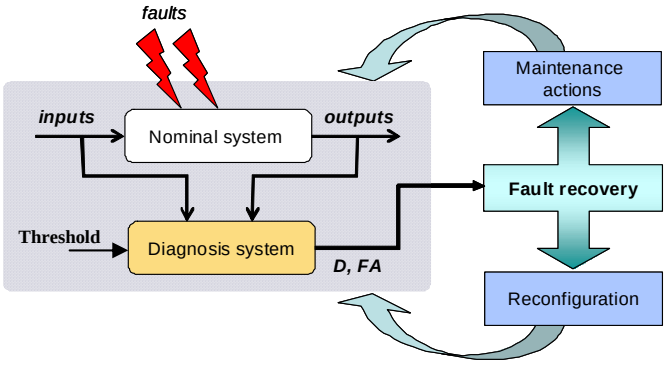

Fig. 2. The principle of fault-tolerance architectures.

A. The fault detection and isolation (FDI)

The diagnosis or *FDI* feature is a key function in fault tolerant systems. Indeed, the diagnosis system allows the detection of faults/failures occurrence on the system's components as well as their localization. This is essential to undertake some recovery actions as reconfiguration and maintenance (Fig. 2).

There are many procedures to perform fault diagnosis [12] and one common way is to use a fault-free model of the system, like its observer, to calculate some system's variables when it is not faulty. These variables are compared to the real ones measured on the real system. The difference between these variables is called the *residual r*. Since the systems are generally operating in noisy environments, the residual is compared to some threshold *J* to make decision: an alarm will be produced if $r > J$. According to the value of threshold parameter and to the fault nature, a fault may be detected correctly (*D*), missed detected (*MD*). Also, an alarm may be produced while there is no fault and this is called a false alarm (*FA*). The probabilities of *MD* and *FA* constitute a measure of diagnosis performances, which have an impact on the actions to be undertaken. Thus, they should be considered in the dependability analysis problem of the FT-system [3].

B. Reconfiguration

This function allows to reconfigure either the material architecture of the system or the control policy when an alarm is produced by the diagnosis procedure. Material reconfiguraion consists in using secondary components, called backups, used in redundancy with the main components of the nominal system. When an alarm is produced, the system swiches to the backup system/component.

When the system is automated, the controller may be designed to be robust to some faults or many control laws are established to deal with some faults, and the system may swich from one control law to another one to minimise the fault's effect on the FT system.

C. Maintenance/component replacement

When an alarm is produced and the system swiches from one supposed faulty component to its backup, a repair action or replacement action could be undertaken on the faulty component. This allows to swich back to the main component for example and to improve the system's dependability.

IV. SANs Based Modeling for Performances Analysis

In, [3] we proposed an integrated modeling approach to design SAN models for the availability assessment purpose. This approach considers the explicit modeling of diagnosis performances, redundancy policies and maintenance actions in the simulation model. This approach will be recalled on an application example and extensions of it are presented to make the model suitable to assess the reliability factor as well as the maintenance costs.

A. Description of the studied process

Let's consider a tank for heating and controlling the flow of a liquid (Fig. 3). It gets at its entry a liquid characterized by a flow rate Q_e and a temperature T_e. The electric power $P(t)$ delivered by two resistors R_1 and R_2 is used for heating the liquid. One controller acts on the resistors to control the liquid

temperature T_o while the other acts on the valve to maintain the output flow Q_o constant. Three sensors are used: *SensorT* to measure the temperature, *SensorF* to measure the flow rate and *SensorH* for the height measure. As the flow rate can be established from the liquid height, these two latter sensors are considered to be in redundancy and the flow control loop can use both of them. *SensorF* is monitored by a diagnosis system to allow reconfiguration to *SensorH* when an alarm occurs.

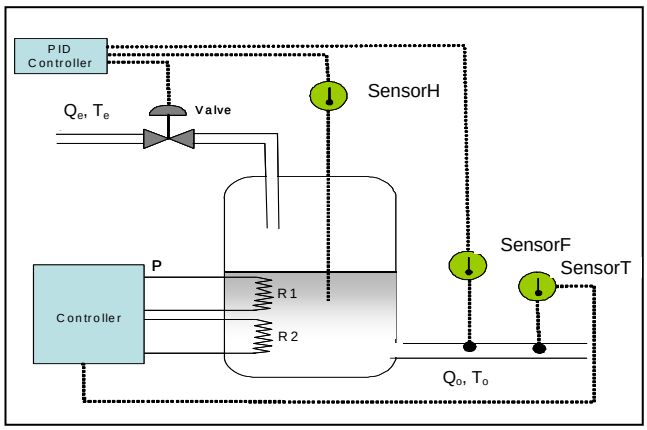

Fig. 3. The water heating process description.

B. Description of the process modeling using SANs

The considered process has two main functions: flow control and temperature control. A functional analysis has been conducted according to these two main functions as well as the failure analysis. The basic idea of this analysis is to consider that a control function is lost whether the controller, sensors or actuators are failed (Fig. 4). This study is focusing on the flow loop control part since it includes the fault tolerance functions described before.

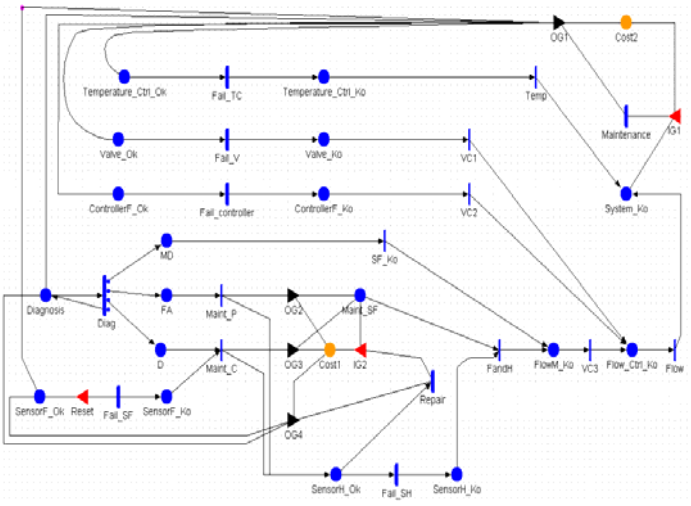

Fig. 4. The SAN model of the water heating process for maintenance cost and availability assessment (model *P&M*).

1) *Components modeling:* Each physical component C_j of the system can be modelled by two places: $\{C_j\_Ok, C_j\_Ko\}$ where a token on a place C_j_Ok (resp. C_j_Ko) means the component C_j is up (resp. failed or down). The marking of these places satisfies the inequality: $M(C_j\_Ok) + M(C_j\_Ko) \le 1$. For example *SensorF* is modelled by two places *SensorF_Ok* and *SensorF_Ko* related by a timed activity *Fail_SF*. It has an exponentially distributed duration with parameter $\lambda_{SensorF}$ which is the failure rate of the sensor. The same reasoning is applied for all the system's components. Notice also that the initial marking of place *SensorH_Ok* depends on how the backup sensor *SensorH* is used: $M_0(SensorH\_Ok) = 0$ for passive redundancy and $M_0(SensorH\_Ok) = 1$ for active redundancy.

2) *Diagnosis modeling:* the diagnosis system is considered explicitly in terms of its performances. It is modeled as a three events generator: *D*, *FA* and *MD* [3]. Knowing the probability of these events (resp. P_D, P_{FA} and P_{MD}), the diagnosis system can be modeled by four places: *Diagnosis* which has an initial marking of one, and places *D*, *FA* and *MD* modeling the subsequent events of the same name. These last three places are related to place *Diagnosis* through a deterministic timed activity *Diag* with four cases probabilities, such that:

$P(case1) = P_{MD}$ *if* $M(SensorF\_Ok) = 0$ and 0 otherwise,

$P(case2) = P_{FA}$ *if* $M(SensorF\_Ok) = 1$ and 0 otherwise,

$P(case3) = P_D$ *if* $M(SensorF\_Ok) = 0$ and 0 otherwise,

$P(case4) = 1 - (P_D + P_{MD})$ *if* $M(SensorF\_Ok) = 0$ and $(1 - P_{FA})$ otherwise.

Place *MD* enables the activity *SF_Ko* which adds a token in place *FlowM_Ko*. This latter models the failure of flow measure part. Place *D* (resp. *FA*) models a diagnosis produced alarm and enables the activity *Maint_C* (resp. *Maint_P*) which models a corrective (resp. preventive) maintenance action request.

3) *Other submodels*:

- Place *Maint_SF* with its output timed activity *Repair* model the maintenance actions on *SensorF*.
- Activity *FandH* has two input places: *SensorH_Ko* and *Maint_SF* modeling respectively the fact that the backup sensor is down and the principal sensor is turned-off for maintenance. It allows adding a token on place *FlowM_Ko* previously described. Notice that activity *FandH* plays the role of an AND operator.
- The flow control part is down (place *Flow_Ctrl_Ko*) if the actuators or the controller or the sensors are down, modeled by activities VC_1, VC_2 and VC_3 respectively.
- The whole system is down (place *System_Ko*) if the temperature control system (activity *Temp*) or the flow control system (activity *Flow_Ctrl_Ko*) is down.
- The input gate *Reset* is used to prevent *SensorF* from falling down (completion of activity *Fail_SF*) when it is turned-off for maintenance purpose.
- If *SensorH* is used in passive redundancy, then place *SensorH_Ok* is connected to activities *Maint_C* and *Maint_P*. A token will be added in it when an alarm is produced. If active redundancy in employed, place

SensorH_Ok will not be connected to these activities and will be initially marked.

- The maintenance on the whole system is modeled thanks to the timed activity *Maintenance*.
- *Cost1* and *Cost2* are two extended places that allow the calculation of some performance amounts like maintenance cost. *Cost1* is associated to a 2-dimension array, *X*, to compute the cost induced by a maintenance/inspection action when a false alarm occurs (component *X*(1)) or when a correct alarm is produced (component *X*(2)). Each time the activity *Maint_P* (resp. *Maint_C*) completes, the cost *X*(1) (resp. *X*(2)) is calculated according to $X(1)=X(1)+T_p*C_P$ (resp. $X(2)=X(2)+T_c*C_c$) thanks to the output function of the output gate *OG2* (resp. *OG3*). C_c and C_p are time unit costs for respectively corrective and preventive sensor's maintenance action. T_c and T_p are respectively the repair duration for a corrective and preventive maintenance action on the monitored sensor. These durations can be constant or can be randomly generated and affected to the activity *Repair*. In the last case, the extended place *Cost1* and the output gates *OG2*, *OG3* and *OG4* are used to define and reset this duration. *Cost1* should then define a 4-dimension array, *X*, where $X(3)=T_c$ and $X(4)=T_p$. The input gate *IG2* allows the reading of these quantities (i.e., *X*(3) and *X*(4)) and to affect them to the timed activity *Repair*. The exact same reasoning can be applied to place *Cost2* affected to the variable *y* denoting the cost of a global maintenance on the system. The output gate *OG1* computes this cost according to $y=y+T_T*C_T$, where T_T denotes the repair time of the system and C_T is the time unit repair cost (with $C_T>C_p$ and $C_T>C_c$).
- The model discussed in this section considers that the system is repairable and allows the availability assessment. If the reliability factor is considered, the activity *Maintenance* should be deleted as well as its input and output arcs, so that place *System_Ko* becomes an absorbing place. The way the reliability will be computed is explained in the next section.

V. Simulation Study and Results

The modelling approach explained in the previous section is combined with *Monte Carlo* (*MC*) simulation in order to assess some performance amounts. Each developed SAN model will be executed multiple times using different randomly generated event streams. Each execution generates a different trajectory through the possible event space of the system, called history. It is necessary to generate many histories to get statistically significant estimations. All the SAN models used in this paper are developed using the *Mobïus* software tool. The simulation's stopping criterion can be either the number of simulated histories, N_h, or the desired *confidence level*. The confidence level specifies the desired probability that the exact value of the measured variable will be within the specified interval around the variable estimate.

In this study, simulations are conducted over at least 5.10^4 and at most 8.10^5 histories. The simulator can also stop if 95% of the results are contained within an interval of 5% around their mean value. The system's parameters like the components' failure rates and the parameters of the timed activities are given in table 1.

TABLE I. Parameters of the distribution functions associated to the timed activities

Exponential distribution (λ in $t.u^{-1}$)					**Uniform distribution (*A* et *B* in *t.u*)**			
λ_{Temp}	λ_{Valve}	λ_{CtrlF}	$\lambda_{SensorF}$	$\lambda_{SensorH}$	A_1	B_1	A_2	B_2
2.10^{-4}	5.10^{-4}	3.10^{-4}	5.10^{-3}	5.10^{-3}	10	25	20	100

Parameters A_1 and B_1 (resp. A_2 and B_2) are used for the timed activity *Repair* (resp. *Maintenance*). Notice that for the reliability evaluation model, the activity *Maintenance* doesn't exist. Also, the repair action when a false alarm occurs can be seen as a preventive maintenance action or simply an inspection action.

The *MC* simulations are conducted for various diagnosis performance factors to show their impact on the overall performances.

The major advantage of our SAN-modelling approach is its algorithm complexity, which is polynomial, which makes it more attractive over some modelling approaches based on automata for example. Consequently, the simulation running time is very short (only few seconds, on an *Intel® core™ Duo* CPU with a clock speed of 2.26 Ghz).

A. Reliability assessment

Four models (R_k, k=1, 4) are derived using the previous procedure to study the impact of the diagnosis performances, maintenance and redundancy policies on the system's reliability factor. In models R_1 and R_2 the backup sensor is in passive redundancy and *SensorF* is maintained in R_1 but not in R_2. In R_3 and R_4 the backup sensor is in active redundancy and *SensorF* is maintained only in R_3.

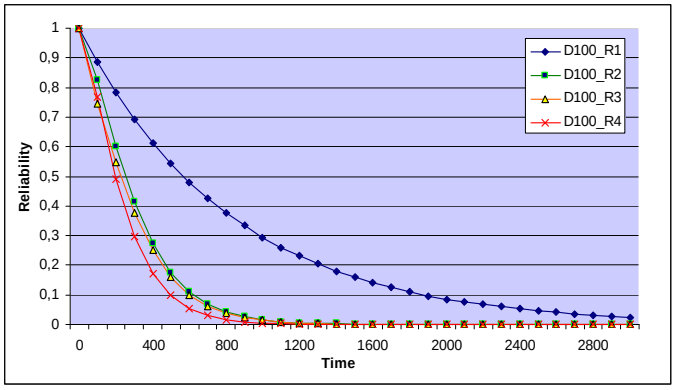

Fig. 5. The time evolution of the FT-system reliability when the fault detection is made with certainty (P_D=100%) for the models R_k, k=1,4.

If for N_h simulated histories, $N_{ok}(t)$ is the number of times that the place *System_Ko* got an empty marking at the instant

of time t, then the system's reliability by that time, $R(t)$, is computed as: $R(t) = N_{Ok}(t) / N_h$.

The history duration is fixed to T_h=3000 *t.u*, since after this date, the reliability is almost null.

When the fault detection is made with certainty (P_D=100%) the model *R1* (i.e. the one where *SensorF* is maintained and reconfigured to its passive backup *SensorH*) is the one which gives the best system's reliability. This shows the impact of these maintenance and redundancy policies. The worst reliability is obtained for model *R4*, where no recovery actions are undertaken. The system's reliability for model *R2* is quite similar to the one of model *R3* but better. This tendency changes when the diagnosis system is not perfect. For example when P_{FA}=10% and P_D=80%, model *R3* (when active redundancy is used) gives a better reliability than *R2* and *R4* (Fig. 6). This can be explained by the fact that for model *R2* the false alarm will make the system reconfigure from the monitored sensor to its backup while it is not failed. This makes the whole system more vulnerable than when the two sensors are both used (model *R3*). In fact model *R2* is more sensitive to the diagnosis performances than model *R3*.

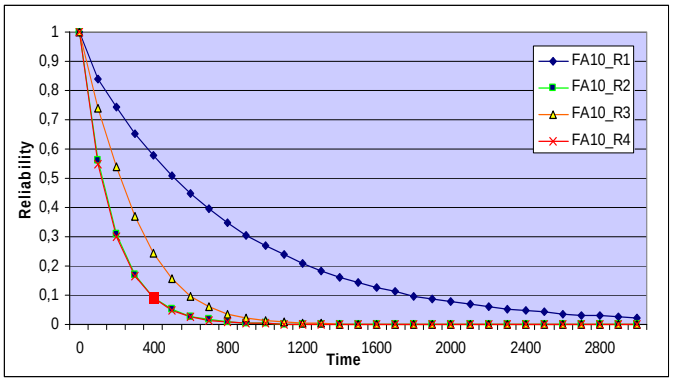

Fig. 6. The time evolution of the FT-system reliability when the diagnosis is not perfect (P_{FA}=10% and P_D=80%) for the simulation models R_k, k=1,4.

Notice that, for all the four models R_k (k=1, 4), the system's reliability is better when the detection is made with certainty (i.e. P_D=100%). For example for model *R1*, when t=400 *t.u*, the system's reliability is of 0.614 when P_D=100% while it is of 0.57 when P_{FA}=10%.

B. Maintenance costs assessment

Three SAN-models are designed to make this study:

- *P&M model*: the monitored sensor, *SensorF*, is maintained when an alarm is produced and its backup, *SensorH*, is in passive redundancy.
- *P&NM model*: passive redundancy is employed and no maintenance is made on supervised sensor.
- *NoDiag model*: there is no diagnosis system in the system which means there are no maintenance and reconfiguration actions on the system.

The time unit cost affected to maintenance of the supervised sensor is C_c=0.75 for a corrective maintenance and C_p=0.1 for a preventive maintenance. While the time unit cost affected to the maintenance of the whole system when failed is C_T=3.

For these simulations, the history duration is fixed to T_h=20000 *t.u*.

The total maintenance cost is calculated as a sum of the marking of the two extended places *Cost1* and *Cost2*. This cost is computed for various false alarm rates (Fig. 7). Notice that the case where P_{FA}=0 corresponds to the case where the detection is made with certainty P_D=100%.

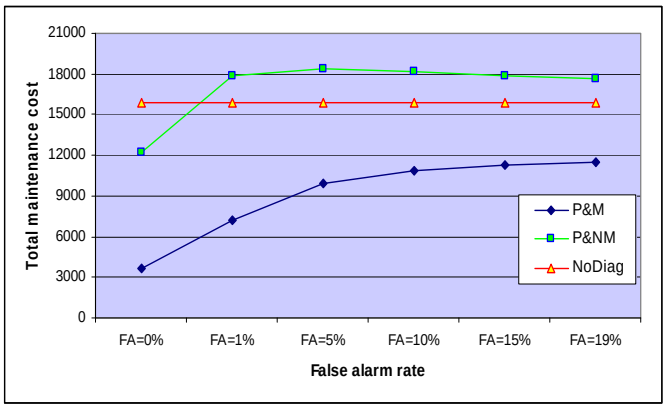

Fig. 7. The diagnosis performance impact over the total maintenance cost.

From these simulations results it can be seen that the total cost for model "*P&NM*" is approximately stable when the false alarm rate increases, unlike the model "*P&M*". In fact, the total maintenance cost increases with the *FA* rate, which is predictable since the monitored sensor is maintained each time a false alarm occurs. (Fig. 8) shows the contribution of the preventive maintenance action on the total maintenance cost. Indeed, the preventive maintenance cost is proportional to the total number of times that *SensorF* is maintained which increases with the *FA* rate. (Fig. 8) shows how the corrective maintenance cost of the sensor decreases in favor of its preventive maintenance cost as the *FA* rate increases. Indeed, *SensorF* is unable to fail for real since it is constantly maintained because of the false alarms.

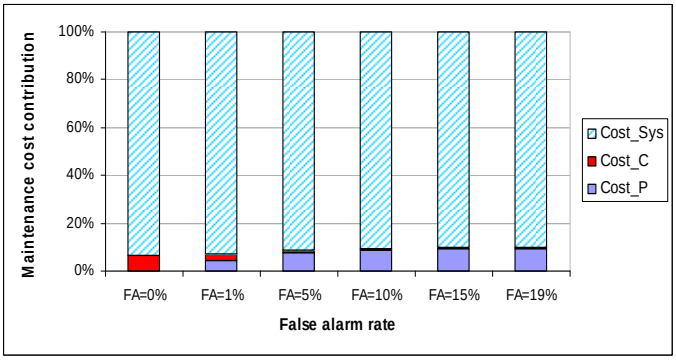

Fig. 8. The sensor's maintenance cost contribution to the total maintenance cost (*Cost_P* and *Cost_C* are preventive and corrective maintenance costs of the sensor and *Cost_Sys* is the system's repairing cost when failed) .

Notice also in (fig. 7) that despite the fact that the cost increases with the *FA* rate for the model "*P&M*", it remains less important that the one of "*P&NM*" and "*NoDiag*" models. This can be explained by the fact that the mean availability of the system in the case "*P&M*" is the best (Fig. 9). From P_{FA}=1% and on, the system's availability of model "*P&NM*" is worst than the case where no diagnosis system is employed (model "*NoDiag*"). This explains why the total maintenance cost is also higher. This also means that with such diagnosis performances and system's parameters, it is better not to implement a diagnosis system which produces alarms excessively without improving the system's availability while increasing unnecessarily the system's maintenance cost.

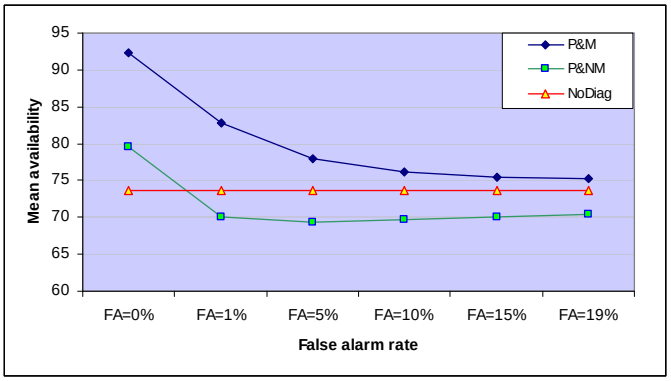

Fig. 9. The mean availability evolution according to the *FA* rate.

For all the simulations results of this section, notice that the usually made hypothesis about perfect fault detection (*i.e.* P_D=100%) is too optimistic since it is the one which always gives the best results.

VI. Conclusion

This paper proposes to use stochastic activity networks to model fault tolerant systems to study the impact of some functions on the system performances. This modeling approach is combined with Monte Carlo simulation to assess some RAMS factors like the reliability, availability and the cost due the maintenance actions.

Indeed, according to the diagnosis made decisions, various actions can be undertaken to cover the eventual faults/failures, like the switching to redundant components and the replacement or repairing of the faulty components. All these functions interact with each others in order to make the whole system more dependable. The question one can ask is about the real efficiency of these functions and their impact on the system's performances: is the dependability really increased? If yes, by how much?

The paper shows the necessity of an integrated modeling approach to make such an analysis. It proposes a simulation study to show how the reliability of the fault tolerant system behaves according to the diagnosis parameters, maintenance policy (repairing/replacing the monitored component or not) and redundancy policy (backup component is in passive or active redundancy). Another simulation study shows the impact of the diagnosis system as well as the maintenance policy on the total maintenance cost.

The advantage of this approach and the chosen modeling formalism, in comparison with others approaches and formalisms, is the algorithm complexity according to the system's size.

One perspective of this work is to consider the modeling of more complex reconfiguration policies, control laws and diagnosis tuning parameters and to study their impact on the system's dependability factors.

References

[1] F. Guenab, W. Schön and J-L. Boulanger, "Système tolérant aux défauts : Synthèse d'une méthode de reconfiguration et/ou restructuration intégrant la fiabilité de certains composants" Journal Européen des Systèmes Automatisés, vol. 43, 2009, pp. 1149-1178.

[2] Aslund, J. Biteus, J. Frisk, E. Krysander, M. & Nielson, N. 2007. "Safety analysis of autonomous systems by extended fault tree analysis", International Journal of Adaptive Control and Signal Processing, 21, pp.287-298.

[3] S. Maza, "Dynamic modeling and simulation of fault tolerant systems based on stochastic activity networks," in Proc IMechE Part O: Journal of Risk and Reliability, vol. 226, 2012, pp. 455-463.

[4] S. Maza and J.F. Petin, "On the use of stochastic activity networks to assess the availability of a fault-tolerant system", in Lambda Mu conference, Tours 2012, France..

[5] G.A. Perez Castaneda, J-F. Aubry and N. Brinzei, "Stochastic hybrid automata model for dynamic reliability assessment", in Proc IMechE Part O: Journal of Risk and Reliability, vol. 225, 2011, pp. 28-41.

[6] A. Cabarbaye and R. Laulheret, " Evaluation de la sûreté de fonctionnement des systèmes dynamiques par modélisation recursive", 6ème congrès international pluridisciplinaire, qualité et sûreté de fonctionnement, *Qualita 2005*, Bordeaux.

[7] M.A. Marsan, "Stochastic Petri Nets: an Elementary Introduction", Springer Verlag, 'Lecture Notes in Computer Science', 1990.

[8] G. Chiola and M.A. Marsan, "Generalized stochastic Petri nets: a definition at the net level and its implications", IEEE Transactions on Software Enfineering, Vol. 19, 1993, pp. 89-107.

[9] A. Mogavar and J.F. Meyer, "Performability modeling with stochastic activity network", Proceeding of real-time systems symposium, Austin TX, USA, 1984, pp. 215-224.

[10] W.H. Sanders and J.F. Meyer, "Stochastic activity networks: formal definitions and concepts", Lectures on formal methods and performance analysis. First EEF/Euro summer school on trends in computer science. Springer-Verlag New York, 2002, pp. 315-343.

[11] R. David, H. Alla, "Petri Nets and Grafcet: Tools for Modelling Discrete Event Systems", Prentice Hall, 1992.

[12] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems", Annual Reviews in Control, Vol. 32, 2008, pp.229-252.