Dynamic modelling and simulation of fault-tolerant systems based on stochastic activity networks

Samia Maza

Dynamic modelling and simulation of fault tolerant systems based on Stochastic Activity Networks

Samia Maza
Centre de Recherche en Automatique de Nancy, Nancy-Université Institut National Polytechnique de Lorraine,
2 avenue de la Forêt de Haye, 54516 Vandoeuvre Lès Nancy, France.
Email : samia.maza@ensem.inpl-nancy.fr

Abstract: Dependability analysis is crucial to control the risks resulting from failures in modern industrial systems whose complexity increases by leaps and bounds. This paper proposes a modeling approach to construct dynamic models of fault-tolerant (FT) systems based on Stochastic Activity Networks (SANs). This approach allows the systematic inclusion of the diagnosis performances to make the dependability analysis. This SAN-model is used jointly with the Monte Carlo simulation to make a study of the impact of diagnosis' performances on the availability of a FT system when various redundancy and maintenance policies are employed.

Keywords: Stochastic Actvitiy Networks (SANs), dependability, availability, diagnosis, Monte Carlo simulation, fault-tolerant systems, maintenance.

1. Introduction

To meet the productivity and safety requirements in industrial systems, many components are added to the original system or process in order to improve its dependability, such as supervision or diagnosis systems, control systems and reconfiguration or backup systems. This leads to the creation of autonomous and adaptive systems capable of making decisions in a given context. These systems become more and more complex. Consequently, the dependability analysis in those systems becomes a difficult task. The overall system, *i.e.* the one composed of the process and its supervision and backup subsystems, is called a fault-tolerant (FT) system. The role of the supervision system is to diagnose the occurrence of faults, *i.e.*, to detect and localize the system's faults. However, the backup system allows the reconfiguration

when faults occur. These components are used to improve the system's reliability. Nevertheless, they are not totally reliable and therefore, their performances should be taken into account when assessing the dependability of the FT system.

In fact, fault detection is based on some diagnosis algorithm which defines a procedure to detect a failure based on some tuning parameters. The quality of detection depends on those parameters, and so are the actions needed to recover from faults like reconfiguration and maintenance. This shows that the performance of the diagnosis systems should be considered explicitly when evaluating the system's dependability. In the same way, the dependability information and objectives could be considered in fault detection and isolation (*FDI*) procedures, to improve the decision making.

In other words, the diagnosis issue and dependability analysis should be considered jointly in order to improve the system's performances.

There are few papers in the literature which deal with the interaction between supervision and dependability analysis and design. For example, in [1] the authors consider the sensor placement problem by combining a fault diagnostic observability study by signed directed graphs and reliability information on component failures probability. Weber *et al* [2] propose a new approach that improves the performance of the decision making in fault diagnosis by taking into account a priori knowledge of the system/components' reliability. Aslund *et al* [3], consider the safety study of FT control systems that include diagnosis subsystem. They propose an approach allowing the inclusion of diagnosis performance in the fault-tree analysis in order to evaluate its impact on the overall system's safety. In the same way, Gustafsson *et al*, propose a method to optimize the detection threshold based on the previously cited approach [4]. Bonivento *et al* [5] suggest a procedure for evaluating reliability of diagnostic systems in terms of capability of not generating false alarms and missed diagnosis using statistical tools. Castaneda *et al* [6] address the problem of dynamic reliability estimation of hybrid systems modelled by stochastic hybrid Automata. Some diagnosis performances are included in their simulation study. Guenab *et al* [7]

work on FT control systems and their reconfiguration. They propose a control strategy which incorporates both the reliability and dynamic performance of the system for control reconfiguration.

The aim of this paper is, in one side to propose a modelling approach which systematically includes the diagnosis system, to make the dependability analysis of FT systems. For that, stochastic activity networks (SANs) are used. The major advantage of such formalism (i.e. SANs) is that it allows the modelling of dynamic systems by modelling all their possible states, but unlike tools such as automata and Markov processes, this can be done simply and in a compact manner. On the other side, the proposed modelling approach is combined with Monte Carlo simulation to assess the system's availability and to study the impact of the supervisor performances. The use of simulation is justified by the fact that the considered systems may have non-homogenous components; active and passive redundancies, repairable components, they may include both discrete and continuous dynamics, etc. This makes the analytical formulation very difficult. The paper deals with repairable FT systems and is organized as follows:

Section 2 is dedicated to the dependability analysis tools and presents a description of the stochastic activity networks in comparison to the well known Petri nets (*PNs*). The principle of fault detection in diagnosis systems and some of its performances are discussed in section 3. Section 4 is devoted to the presentation the proposed modelling approach. This procedure is applied on an automated thermal process in section 5. This section is also dedicated to the simulation study and results discussion. Finally, the paper is concluded in section 6 where some perspectives are also given.

2. Dependability analysis

The dependability of a system can be defined as a property that allows its users to have a justified reliance on the service they are delivered. It is described by various non-functional properties such as: reliability, availability, safety and maintenability [8]. The present paper deals essentially with the availability factor. Availability analysis is performed to verify that an item has a satisfactory probability of being

operational, so it can achieve its intended objective [9]. Formally, it is the probability that a system, under stated conditions, is operational at a given time.

There is a variety of classical methods for reliability and availability analysis, which can be either static or dynamic, like fault-trees, Markov processes and Petri nets [10]. This section presents the stochastic activity networks (SANs). SANs are extension of Petri nets (PNs) [11]. A Petri net structure is a directed weighted bipartite graph defined by a 4-*tuple PN*=(*P, T, Pre, Post*), where *T* and *P* are two distinct sets of vertexes. $T=\{t_1, t_2,\ldots t_n\}$ is a set of transitions, and $P=\{p_1, p_2, \ldots p_k\}$ 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. *Pre* and *Post* are two applications, defined from the set of arcs to the set of natural numbers $\mathbb{N}$: $pre(p_i, T_j): P \times T \to \mathbb{N}$ and $post(T_i, p_j): T \times P \to \mathbb{N}$. They define the valuation of arcs relating places to transitions (*Pre*) and transitions to places (*Post*). A marked Petri net is a 5-*tuple* PN_m=(*P, T, Pre, Post,* M_0), where M_0 is the initial marking of the *PN*. It is a *k*-dimension vector, where *k* is the number of places and models the system's initial state. A marking vector could be written as: $M = (M(p_1), M(p_2), \ldots M(p_k))^T$, where $M(p_i)$ is the number of tokens in place p_i.

The SANs were first introduced by Mogavar *et al*. [12], and used as a modelling formalism for the performance and dependability evaluation of a wide range of systems [13]. Hereafter is given an informal definition of SANs in comparison to PNs.

- *Places*, as for Petri nets, can be seen as a state of the modelled system, and are represented graphically by circles (Fig. 1).
- *Activities*, like *transitions* in Petri nets, they are of two kinds: timed and instantaneous. This duration can be either deterministic or stochastic. In Petri nets vocabulary, we say that transitions fire while in SANs vocabulary, activities complete. Each activity has a non-zero integer number of *cases probabilities*.

- *Cases probabilities* allow the modelling of the uncertainty about the enabled activity to complete. Here, the term *case* is used to denote a possible action that may be taken upon the completion of an event such as a routing choice in a network, or a failure mode in a faulty system. *Cases* are graphically represented by small circles on the right side of an activity (Fig. 1). Moreover, a *case* probability distribution can depend on the marking of the network when the activity completes.
- *Input gates* are used to control the activation or enabling of the activities. An input gate is defined by two functions: the *predicate* and the *input function*. The *enabling predicate* of an input gate defines the condition which enables or activates the activity. It depends on the marking of the gate's input places. An activity is enabled when the *predicates* of all its input gates are true. When the activity completes, the new marking of its input places is defined by the gate's *input function*. Input gates are graphically represented by triangles, with the flat side inside connected to the activity via its input arc (Fig. 1).
- *Output gates* are used to change the state of the system when an activity completes. It defines the marking change of the output places thanks to the *output function*. They are graphically represented by triangles, with the flat side inside connected to the activity via its output arc. Each input or output gate is connected to only one activity.

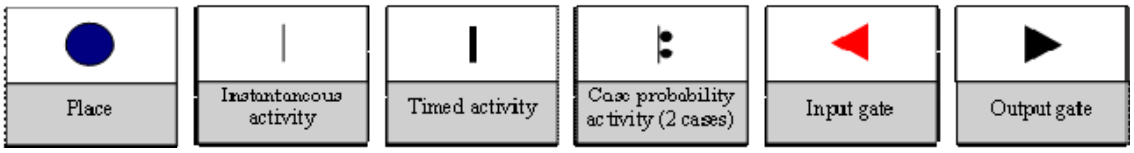


Fig. 1 The graphical representation of SANs elements.

An example of a SAN model with the previously cited elements is given in (Fig. 2). The predicate function of the input gate *I_G* expresses the enabling condition of the timed activity *Timed_A*. This latter is enabled if and only if $M(P1)=4$ and $M(P2)=2$. When this predicate is true, the activity *Timed_A* will

complete after a delay *T* which is the time duration of the activity. The input function of *I_G* specifies the new marking of the places *P1* and *P2* after the completion of *Timed_A*, in this example, *M*(*P1*)= *M*(*P1*)-2 and *M*(*P2*)= *M*(*P2*)-1. The output function of the output gate *O_G* defines the marking of place *P3* after the completion of *Timed_A*, here: *M*(*P3*)=2**M*(*P3*). The instantaneous activity I_A has three cases probabilities, each related to one place. These probabilities are fixed in this example to *P*(*Case1*)=0.3, *P*(*Case2*)=0.2 and *P*(*Case3*)=0.5. This means that *I_A* has for example 50% chances to complete through *case3* and a token will be added to place *P2*. As said before, these probabilities can depend on the marking of some places like *P3*, which shows the modelling power of the SANs.

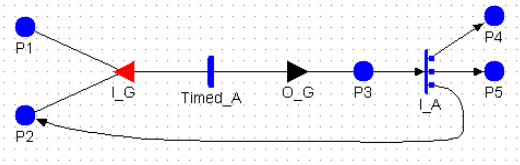


Fig. 2 Example of a SAN model with its previously defined elements.

3. Diagnosis analysis: performance measures

The diagnosis system is a key component in fault-tolerant systems. It allows the detection of faults or abnormal functioning of components under supervision. In FT systems, the fault detection and isolation (FDI) allows reactivity, such as reconfiguration, to avoid losing the system's function, safety, etc. Consequently, fault detection and isolation (*FDI*) procedures are essential to improve system's dependability and different approaches are proposed in the literature to design such procedures (see for example [14]).

One common way to perform fault detection is to define a set of tests quantities r_i, called residuals. A residual is defined as the difference between the measured value of some system's variables and the expected ones, estimated from a system's fault-free model as the observer. In fault-free situations, these residuals should be equal to zero and non-zero otherwise. In practice, systems operate in noisy environment. This may affect the residuals and thus, the decision making. Consequently, the

residuals r_i are compared to some tuning parameter called the thresholds J_i instead of zero. If $r_i > J_i$ then the test is said to alarm.

In the residual evaluation problem and according to the statistical theory: the hypothesis "*component i is Ok*" is called the *null hypothesis* of a test and is denoted H_i^0 [2, 3]. When the supervised component is *Ok* and the test produces an alarm, is called *false alarm* (*FA*). Not alarm when the supervised component is down is called *missed detection* (*MD*). As a conclusion, the residual analysis will produce, according to whether the null hypothesis is true or not, three results: good detection (D), false alarm (FA) and missed detection (MD) as shown in table 1.

	H_i^0 is true	H_i^0 is false
H_i^0 is accepted: $r_i \le J_i$	Ok	MD_i
H_i^0 is rejected: $r_i > J_i$	FA_i	D_i

Table 1. Definition of the diagnosis decisions D_i, FA_i and MD_i

The probability of the events FA_i and MD_i is:

$$P(FA_i) = P(r_i > J_i \,|\, H_i^0 \; true) \text{ and } P(MD_i) = P(r_i \le J_i \,|\, H_i^0 \; false)$$

These probabilities represent measures of the diagnosis performances and the threshold value adjusts the compromise between a small *FA* probability P_{FA} and *MD* probability P_{MD}.

4. The SAN modelling of diagnosis performances for dependability analysis of FT-systems

In the reliability literature, it is always assumed that the detection is made with certainty however; as seen before, the dependability of a system depends on the quality of the detection. Indeed, FDI procedures have a direct impact on the actions made to recover from faults. The dependability of the system is thus tightly related to the performances of the diagnosis system. This section proposes a procedure to model FT-systems using SANs, with a systematic inclusion of the diagnosis performances.

The objective of such modelling is to make reliability/availability analysis. Only components' failures are considered in the paper.

4.1. SAN-modelling of physical components

Each physical component C_j of the system can be modelled by two places: $\{C_j^{up}, C_j^{down}\}$ where a token on a place C_j^{up} (resp. C_j^{down}) means the component C_j is up (resp. failed or down). The marking of these places (i.e. $M(C_j^{up})$ and $M(C_j^{down})$) satisfies the inequality : $M(C_j^{up}) + M(C_j^{down}) \leq 1$. The two places are linked to each other by a timed activity called "$fail_j$" representing the failure of C_j (Fig. 3). This duration can follow any probabilistic distribution such as the exponential distribution function.

If a component C_j is repairable, maintenance action can be modelled by some timed activity connected to place C_j^{up} to add a token in it when the component is repaired. This activity will not be connected to place C_j^{down} but to the diagnosis SAN sub-model since maintenance is made only if the supervision system diagnoses the failure. Indeed, if the detection is made instantaneously and with certainty, the activity modelling the repair action will connect place C_j^{down} to C_j^{up} as for automata or Markov processes. But as explained before, this repair action depends on the quality of detection.

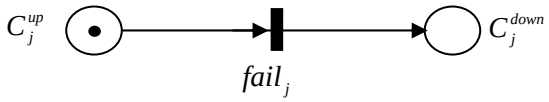


Fig. 3 The SAN modeling of a physical component C_j

4.2. SAN-modelling of a backup system

A backup component/system is modelled as any physical component by two places $\{C_{Backup}^{up}, C_{Backup}^{down}\}$ excepted that the initial marking of these places depends on the redundancy policy:

$$M_0(C^{up}_{Backup}) = \begin{cases} 1, \text{ for hot active redundancy} \\ 0, \text{for cold passive redundancy} \end{cases}.$$

4.3. SAN-modelling of the diagnosis system

The diagnosis system can be modelled as a generator of three mutually exclusive events: D , FA and MD (*Cf.* §3). Knowing the probability of these, the supervisor could be modelled by a place, called $ALGO$, with an initial marking of one token (*i.e.* $M_0(ALGO) = 1$). The place *ALGO* has an output activity, named $Diag$, with four *cases probabilities* (Fig. 4). Each place in { $P_{Detection}$, $P_{MDetection}$, P_{FAlarm} } is related to one $Case_k$ (k=1,3) and models the events *D*, *MD* and *FA* respectively. For a specific network's marking, these cases have the probability of their associated event, *i.e.*, P_{MD}, P_{FA} and P_D respectively. $Case_4$ is added only to satisfy the condition $\sum_{k=1}^{4} P(Case_k) = 1$ (where $P(Case_k)$ is the probability of case k) and means no event is produced by the supervisor. $Case_4$ is connected to the place $ALGO$ to conserve the token in it when $Diag$ completes through this case in order to activate "*Diag*".

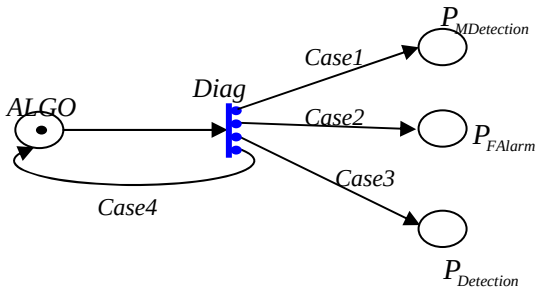


Fig. 4 The SAN-modelling of the diagnosis system and its performance

Notice however, that these probabilities depend also on the markings of the SAN model associated with the supervised component/system. Consider for example that a component C_k is supervised. Then the probability of each *case* is given by the following:

$$P(Case1)=\begin{cases}P_{MD} \ \ if \ \ M(C_k^{down})=1\\ 0 \ \ otherwise\end{cases}, P(Case\,2)=\begin{cases}P_{FA} \ \ if \ \ M(C_k^{up})=1\\ 0 \ \ otherwise\end{cases}, P(Case3)=\begin{cases}P_{D} \ \ if \ \ M(C_k^{down})=1\\ 0 \ \ otherwise\end{cases}$$ and

$$P(Case4)=\begin{cases}1-P_{MD}-P_{D} \ \ if \ \ M(C_k^{down})=1\\ 1-P_{FA} \ \ otherwise\end{cases}$$

The *SAN-model* of the diagnosis system satisfies the following:

$$\begin{cases}M_0(ALGO)=1\\ M(ALGO)+M(P_{MDetection})+M(P_{FAlarm})+M(P_{Detection})\le 1\end{cases}$$

4.4. The global SAN-modelling

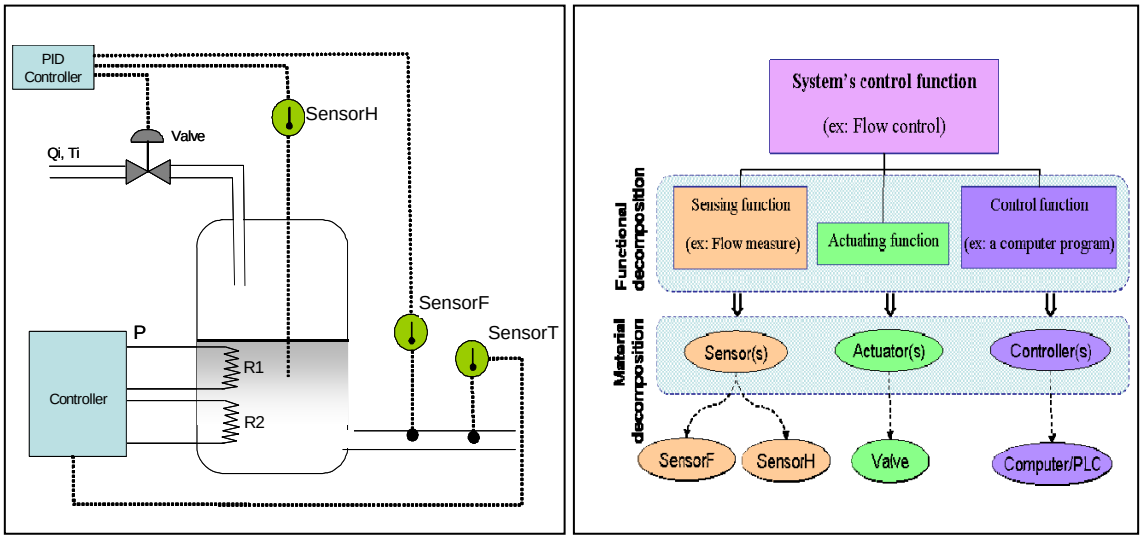
In the final SAN-model, each place in $\{P_{Detection}, P_{MDetection}, P_{FAlarm}\}$ will be related, using logic operators like AND and OR, to activities of other *SAN* sub-models, like the components' models, to activate them. For sake of clarity, the combination of such sub-models will be explained on an example in the following section.

Notice here that the SAN-modelling of the system's components is quite similar to the automata modelling since the places of the components' SAN-models are binary and denote the components' states. But when automata are used, the overall model states number will explode exponentially according to the components number.

5. Case study

This modelling approach is illustrated hereafter on a simulation example of a heating water process (Fig. 5.a). The aim of such a thermal process is to provide water at a given temperature with a constant flow rate. From the automatic point of view, the process's inputs are the water flow rate Q_i and the heater electric power P and the outputs are the water's temperature T and flow rate Q_o. The system has two actuators: the valve and the resistors (*R1* and *R2*), and three sensors to measure the temperature (*SensorT*), the flow rate (*SensorF*) and the water height (*SensorH*). Since there is a physical relationship between the flow rate and the water's level [2], the flow and height sensors are considered in redundancy.

The flow controller can use both of them. A diagnosis system monitors the flow sensor which makes the reconfiguration to the height sensor possible when an alarm is produced.



(a) Heating water process

(b) Example of a functional decomposition of the process according to its flow control function

Fig. 5 The controlled heating water process and an example of its functional analysis.

5.1. The SAN modelling of the process

The process described before is composed of two control subsystems: temperature control and flow rate control systems. The functional decomposition and analysis of this automatic system can be made according to this two main control functions (See [15, 16, 17] for details on functional decomposition and analysis). To achieve any control function, an automated system should be equipped with actuators, sensors and controllers. Communication network and software are not considered in this paper. Thus, if one of these components fails, the automated system will not deliver correctly or not at all the service it was designed for (Fig. 5.b). Based on this idea of functional decomposition, the SAN-model of this automated process can be derived easily as shown on (Fig. 6).

For sake of simplicity, The SAN-model construction will be explained only on the flow rate control subsystem, where the diagnosis acts. The temperature control system is considered as a component with its own failure rate λ_{Temp}. It is modelled in (Fig. 6) by places "*Temperature_Ctrl_Ok*" and "*Temperature_Ctrl_Ko*" while the timed activity "*Fail_TC*" models its failure. According to §4, every component in this SAN-model is modelled by a couple of places: "*Component_Ok*" for the *Up* state and "*Component_Ko*" for the *Down* state.

The supervised component, here the flow sensor *SensorF*, is assumed repairable and can be maintained each time the supervisor alarms. This action is modelled by the timed activity "*Repair*" and place "*Maint_SF*". This latter is connected to the instantaneous activities "*Maint_1*" and "*Maint_2*". The activity "*Maint_1*" can be assimilated to a preventive maintenance action since its input place is "*FAlarm*" which models a false alarm. While "*Maint_2*" has two input places: "*SensorF_Ko*" and "*Detection*" which models the failure detection. Activity "*Maint_2*" is used in an *AND*-logic way (i.e. places "*SensorF_Ko*" and "*Detection*" should be marked together) and can be assimilated to a corrective maintenance action. When the diagnosis system produces an alarm, either correct or false, the FT-system will switch from the flow sensor (*SensorF*) to height sensor (*SensorH*). When this latter is used in cold passive redundancy, the place "*SensorH_Ok*" is connected to activities "*Maint_1*" and "*Maint_2*" and will be marked each time an alarm occurs. The place "*FlowMeasure_Ko*" models the failure of the sensing part in the flow control system. It is connected to activities "*Sensoring_Ko*" and "*SF_Ko*": the first one models the fact that *SensorF* is turned-off for maintenance and its backup *SensorH* is down, and the second one models the missed failure of *SensorF*.

The entire flow control system fails (place "*Flow_Ctrl_Ko*") if the valve or the controller or the sensing loop is down. This is modelled by activities "*VC*1", "*VC*2" and "*VC*3" respectively. These

activities are connected to place“ *Flow _ Ctrl _ Ko* ” according to the logical operator *OR*. In the same way, the FT-system fails (place “ *System _ Ko* ”) if the temperature control system or the flow control system is failed (resp. activities “ *Temp* ” and “ *Flow* ”). Then the whole FT-system can be maintained through the timed activity “ *Maintenance* ” and tokens will be added to all places of type “ *Component _ Ok* ” if and only if their marking is null (i.e. the component is down). This condition is expressed in the output gate “ *OG*1 ”. Since place “ *System _ Ko* ” can receive more than one token, the input gate “ *IG*1 ” is used to avoid making a maintenance more than once at a time. The input function of gate “*IG1*” is: *M*(*System_Ko*)=0.

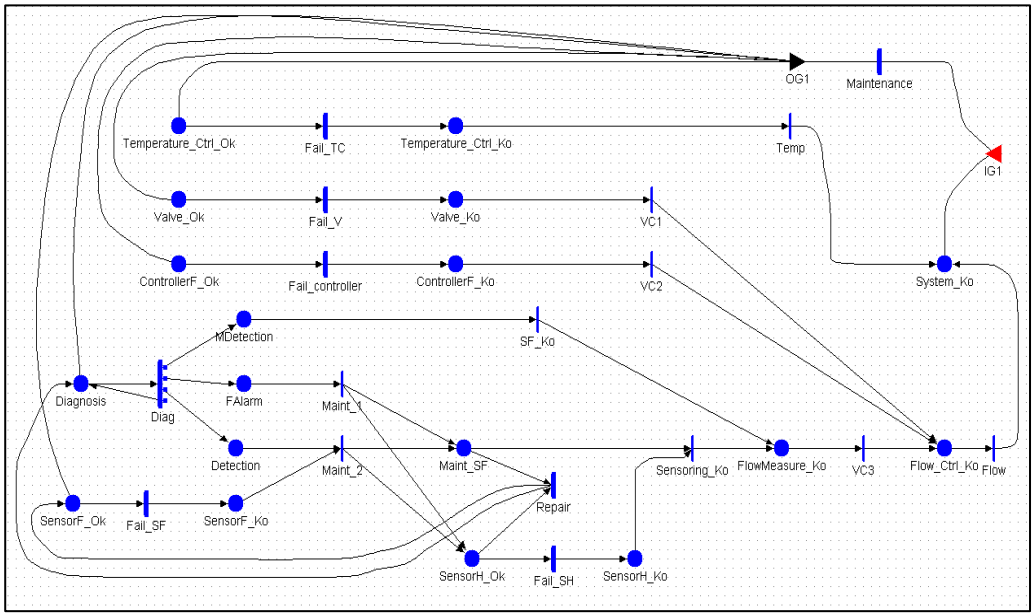


Fig. 6 SAN-model of the water-heating process (*Model B*).

To study the impact of the diagnosis performances and recovery actions on the availability of the supervised process, four simulations SAN-models are designed with various redundancy and maintenance policies, using *Möbius* software tool:

- *Model A (P&NM)*: *SensorH* is in passive cold redundancy and *SensorF* is not maintained;
- *Model B (P&M)*: *SensorH* is in passive cold redundancy and *SensorF* is maintained (Fig. 6);

- *Model C (A&NM)*: *SensorH* is in active redundancy and *SensorF* is not maintained;
- *Model D (A&M)*: *SensorH* is in active redundancy and *SensorF* is maintained;

5.2. Monte Carlo simulation

Monte Carlo simulation is often used to assess the dependability factors estimation. In MC simulation, a model is solved by simulation and is 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. To get statistically significant estimations, it is necessary to generate many trajectories.

Möbius computes confidence intervals as the observations are collected to give an estimate of the accuracy of the calculated estimates. When the desired confidence level for every studied variable is reached, the simulator will stop. 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. However, the *confidence interval* specifies the width of the acceptable interval around the variable estimate. In this study, simulations are conducted over at least 5.10^4 and at most 8.10^5 histories. Each history has a duration T_h of 20000 time units. The simulator stops if the maximum number of histories is reached, or if the discrepancy between the results is less than 5% with a confidence level of 95%, *i.e.* 95% of the results are contained within an interval of 5% around their mean value.

Table 2 shows the values of the timed-activities distribution function parameters used in the simulation study. Here, the failures of components C_j follow an exponential distribution with a constant rate λ_{C_j}.

Exponential distribution					Uniform distribution			
λ_{Temp}	λ_{Valve}	λ_{Ctrl}	$\lambda_{SensorF}$	$\lambda_{SensorH}$	α_1	β_1	α_2	β_2
2.10^{-4}	5.10^{-4}	3.10^{-4}	5.10^{-3}	5.10^{-3}	10	25	20	100

Table 2. Distribution functions of the timed activities.

In the simulation models, it is supposed that a maintenance operator and resources are immediately available when a maintenance action is needed. Its duration is supposed to follow a uniform distribution.

The parameters α_1 and β_1 (resp. α_2 and β_2) denote the lower and upper bound of the uniform distribution associated with the timed activity *Repair* for the supervised component (resp. *Maintenance* for the whole system). The maintenance action may be only an inspection action since false alarms are possible.

5.3. Simulation results

The goal of these simulations is to study the impact of the diagnosis system on the mean availability of the FT-system. Many simulations were conducted with different values of P_{FA}, P_{MD} and P_D. For that, P_D is set to 80% while the values of P_{FA} and P_{MD} are varied between 1% and 19%. The hypothesis of certainty fault detection is also considered (*i.e.* P_D=1). Even if it's false, this is still widely used in the literature.

The statistics on the time occupation of place "*System_Ko*" are used to calculate the mean availablitiy '*A*'.

If T_{System_Ko} is the mean time occupation of place "*System_Ko*", then: $A = (1 - \frac{T_{System\_Ko}}{T_h}) \times 100$.

The simulations were made on an *Intel® core™ Duo* CPU with a clock speed of *2.26 Ghz*. The simulation execution time varies, according to simulation data and model, from few seconds to about 50 minutes.

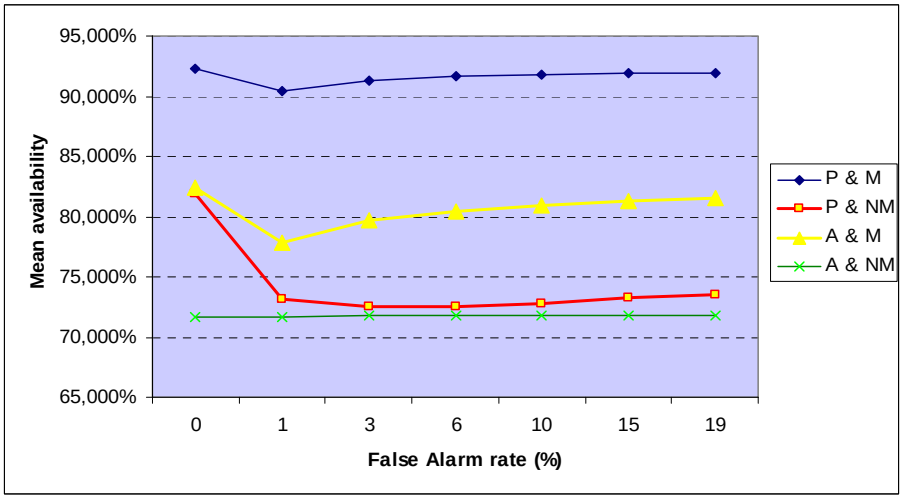


Fig. 7 The mean availability evolution according to the false alarm rate for: *model A* (P & NM), *model B* (P & M), *model C* (A & NM) and *model D* (A & M).

The mean availability results for six different values of P_{FA} and P_{MD} are reported in (Fig. 7). It can be seen that globally for all the models, excepted *model C*, the mean availability increases as the false alarm rate increases. This growth is more significant when maintenance actions are provided to the supervised component (*Models B* and *D*). In addition, these models give better results than *models A* and *C*. This shows that maintaining the supervised component improves the availability of the whole system. This action is possible thanks to the diagnosis.

The *model B*, where passive cold redundancy is employed gives the best mean availability.

Indeed, the backup sensor (*SensorH*) is turned-on only when the supervisor alarms, while the supervised sensor is tuned-off to be maintained. Each time this latter is repaired, the system switches back to the principal sensor. The backup sensor is then tuned-off. Such a redundancy allows the persistence of the liquid's flow measurement part and increases its life-time. However, when the backup sensor is used in active redundancy, even if the information provided by it is not used in the control loop, the sensor is still functioning, and may fail as well as the principal sensor. This is why the *model D* gives a lower availability than *model B*. Indeed, let's consider the contribution in per cent of the flow sensing part, to failure of the system in comparison to the other components of the flow control system (Fig. 8).

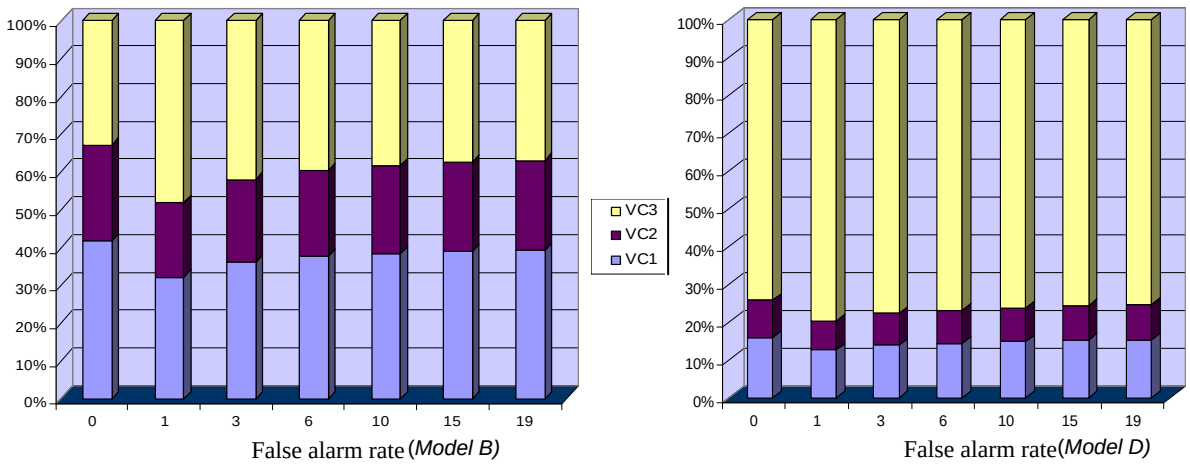


Fig. 8 The cumulative contribution to the failure of the FT-system of the flow sensing part (VC_3) in comparison to the actuating (VC_1) and control (VC_2) parts (*Models B and D*).

Here, VC_j, $(j = 1,3)$ of (Fig. 8) are related to their corresponding activities VCj of (Fig. 6) and express the cumulative contribution of the number of times that these activities have completed leading to the system's failure. The contribution of the flow sensing part, *i.e.* VC_3, is more important when active redundancy is employed (Fig. 8.a) than passive redundancy (Fig. 8.b). For both models *B* and *D*, this contribution decreases as false alarm rate increases.

These results show that the combination of the redundancy policy together with the maintenance policy makes one model better than another. They also show that in the studied system, it is better to over detect faults than to miss them, since the availability is improved as the false alarm probability grows. Indeed, when the supervisor alarms, even if the fault is not real, actions like the reconfiguration and repair of the supervised sensor will increase the system's functioning time. Maintaining a non-faulty component can be assimilated to a preventive maintenance action. Such an action will contribute to reduce the total number of repair actions on the whole system as shown in (Fig. 9). Finally, the hypothesis which considers that failure detection is made with certainty is too optimistic as the system's availability is greater than in any other case. In Figs 7 to 9, it corresponds to the case where the *FA* rate is null.

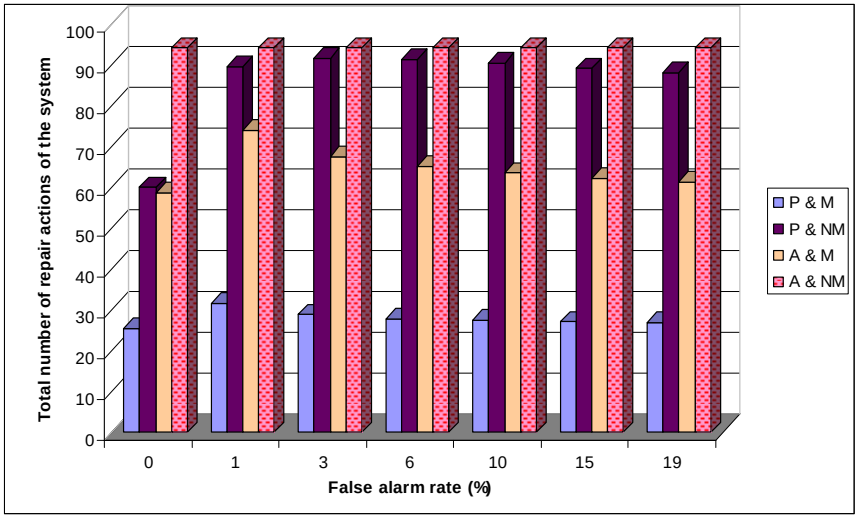


Fig. 9 The total number of repair actions of the FT-system according to the false alarm rate for *model A* (P & NM), *model B* (P & M), *model C* (A & NM) and *model D* (A & M).

6. Conclusion

This paper proposes a method to construct dependability models for fault tolerant systems using the Stochastic Activity Networks. Combined with Monte Carlo simulation, these models allow the evaluation of dependability factors by including events associated with the supervision system and maintenance actions.

This modelling approach was tested on an example of a heating process. A simulation study is made for this FT system, where different redundancy and maintenance policies were employed. Monte-Carlo simulation has been used to evaluate the system's mean availability and to study the impact of diagnosis performances, as well as the redundancy and maintenance policies on the system's availability.

As said in section 3, the diagnosis performances depend on the design parameter called threshold. The next step of this research work is to study the direct impact of this parameter on the system's dependability. For that, there is a need to model the physical process and its supervisor, and to incorporate this diagnosis model into the whole dependability evaluation SAN-model. This may help for choosing the design parameters with respect to the dependability objectives.

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

1. Bhushan, M. & Rengaswamy, R. 2000. Design of sensor location based on various diagnostic observability and reliability. *Computers and Chemical Engineering*, 24, pp.735-741.

2. Weber, P. Theilliol, D. & Aubrun, C. 2008. Component reliability in fault-diagnosis decision making based on dynamic Bayesian networks. *Journal of Risk and Reliability*, 222(2), 161-172.

3. 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.

4. Gustafsson, F. Aslund, J. Frisk, E. Krysander, M. & Nielsen, L. 2008. On threshold optimization in fault-tolerant systems. *IFAC World Congress*, South Korea.

5. Bonivento, C. Capiluppi, M. Marconi, L. Paoli, A. & Rossi, C. 2006. Reliability evaluation for fault diagnosis in complex systems. *Safe Process*.

6. Castaneda, G.A, Aubry, J-F. & Brinzei, N. 2009. Simulation de Monte Carlo par automate stochastique hybride: application à un cas test pour la fiabilité dynamique. *8ème Congrès International pluridisciplinaire Qualita*, Besançon, France.

7. Guenab, F. Schön, W. & Boulanger, J-L. 2009. 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 – No. 10, pp. 1149-1178.

8. Laprie, J.C. 1992. Dependability: Basic concepts and associated terminology. *Dependable Computing and Fault-tolerant Systems*, 5, Springer Verlag.

9. Modarres, M. Kaminskiy, M. & Krivtsov, V. 1999. Reliability engineering and risk analysis, *Marcel Dekker, Inc*.

10. Villemeur, A. 1992. Reliability, Availability, Maintainability and Safety assessment: methods and techniques, *Wiley*.

11. Cassandras, C.G. & Lafortune, S. 1999. *Introduction to discrete event systems*. Kluwer Academic Publishers.

12. Mogavar, A. & Meyer, J.F. 1984. Performability modeling with stochastic activity network. *Proceeding of real-time systems symposium*, p. 215-224, Austin TX, USA.

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

14. Ould Bouamama, B. Samantaray, A.K. Medjaher, K. Staroswieki, M. & Dauphin-Tanguy, G. 2005. Model builder using functional and bond graph tools for FDI deign. *Control Engineering Practice*, vol. 13, pp. 875-891.

15. Conrard, B. Thiriet, J-M & Robert, M. 2005. Distributed system design based on dependability evaluation: a case study on a pilot thermal process. International Journal of Reliability Engineering and System Safety, 88, pp.109-119.

16. Benard, V. Cauffriez, L & Reneaux, D. 2008. The Safe-SADT method for aiding designers to choose and improve dependable architectures for complex automated systems. International Journal of Reliability Engineering and System Safety, 93, pp.179-196.

17. Marca, DA. McGowan, CL. SADT: Structured analysis and design techniques. McGraw-Hill software engineering series; 1988.

LIST OF CAPTIONS

Fig. 1 The graphical representation of SANs elements.

Fig. 2 Example of a SAN model with its previously defined elements.

Fig. 3 The SAN-modelling of a physical component C_j.

Fig. 4 The SAN-modelling of the diagnosis system and its performance.

Fig. 5 The controlled heating water process and an example of its functional analysis.

Fig. 6 SAN model of the water-heating process (*Model B*).

Fig. 7 The mean availability evolution according to the false alarm rate for:

model A (P & NM), *model B* (P & M), *model C* (A & NM) and *model D* (A & M).

Fig. 8 The cumulative contribution to the failure of the FT-system of the flow sensing part (VC_3) in comparison to the actuating (VC_1) and control (VC_2) parts (*Models B and D*).

Fig. 9 The total number of repair actions of the FT-system according to the false alarm rate for *model A* (P & NM), *model B* (P & M), *model C* (A & NM) and *model D* (A & M).

NOTATIONS

AN: Activity Network.

C^{up} : Component *C* is up.

C^{down} : Component *C* is down.

D: Detection.

FA: False Alarm.

FT: Fault-Tolerant

M: Marking of a place/SAN.

MD: Missed Detection.

P_D : fault detection probability (probability of event *D*).

P_{FA} : false alarm probability (probability of event *FA*).

P_{MD} : missed detection probability (probability of event *MD*).

S.A.D.T: Structured Analysis and Design Technique.

SAN: Stochastic Activity Network.