

# DEPENDABILITY EVALUATION OF A SIMPLE MECHATRONIC SYSTEM USING COLOURED PETRI NETS

Gilles MONCELET<sup>\*,\*\*</sup>, Søren CHRISTENSEN<sup>\*\*\*</sup>, Hamid DEMMOU<sup>\*\*</sup>,  
Mario PALUDETTO<sup>\*\*</sup>, José PORRAS<sup>\*</sup>

*\*PSA Peugeot Citroën, 18 rue des Fauvelles, 92256 La Garenne Colombes cedex, France  
Tel: + 33 (0)1 47 69 83 36*

*\*\*LAAS/CNRS, 7 avenue du colonel Roche, 31077 Toulouse cedex, France  
Tel: + 33 (0)5 61 33 62 00, E-mail: moncelet@laas.fr*

*\*\*\*Computer Science Department, Aarhus University, Ny Munkegade 116, DK-8000 Aarhus C, Denmark  
Tel: +45 89 42 32 65, E-mail: schristensen@daimi.aau.dk*

**Abstract:** Mechatronic automotive systems are hybrid systems. Modelling and simulation of the interactions between continuous and discrete parts is essential to evaluate dependability. In this paper we show how a simple mechatronic system can be modelled in the CPN formalism. Quantitative dependability evaluation is obtained thanks to Monte-Carlo simulation. We use the DesignCPN Occurrence Graph tool to validate the model and make a qualitative analysis of the system.

## 1. MOTIVATION

Mechatronic systems mix electric, mechanic, hydraulic and electronic technologies and use a computer control [GUY 94]. Some mechatronic systems like active suspension, automatic gear box, engine control, anti-skating system are already available on today's cars. The aim of the control system is to observe the operative part through physical variables measured by the sensors, and choose the suitable command processed by the actuators. Two kind of actions are possible : continuous or discrete actions. The continuous control process estimate the output error compared to a target value and calculate the new continuous action to reduce the error. A discrete control process detect some event (typically, a threshold overshoot) and choose a new discrete state for the system. A reconfiguration system is a discrete control system dedicated to react against faults of the system components. The architecture of a typical mechatronic system is given by Figure 1. In this article, we deal with discrete control processes only.

In the early design stage of a new mechatronic system, designers have to deal with dependability evaluation [LER 92, HEN 96]. From a functional model, the Preliminary Risk Analysis identifies the events that lead to a catastrophic event, also called « feared events ». The fault tree method is then used for a qualitative and quantitative dependability evaluation.

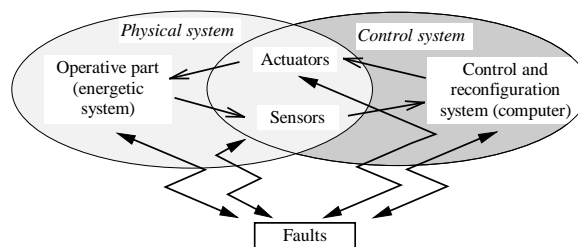


Figure 1: Architecture of mechatronic systems

A fault tree gives the Boolean conditions of occurrence of a feared event. These conditions are written in terms of elementary events, the faults of the basic components of the system [PAG 80]. Efficient algorithms and tools allow

today to compute the feared event occurrence probability given the elementary event failure rates. But, this representation is static and does not take into account reconfigurations.

An alternative to the fault trees is to model the structural and functional interactions between the components of the system in the State Graph formalism [PAG 80]. The modelled states are the operating and fault states of the system. State graphs can describe any kind of finite discrete event system by enumerating the states, but the number of states grows drastically with the number of parallel activities generated by the system. Petri Nets are well suited to model discrete event systems with concurrent and synchronised activities and to cope with the combinatory explosion of the number of states.

For a quantitative dependability evaluation, it is necessary to take into account time as a variable. In a mechatronic system, the delay of state change of a device is captured by associating a delay to a place or a transition in the corresponding Petri Net.

Delays related to repair and fault process are generally modelled by random variables with exponential distribution functions. A Petri Net containing only stochastic time delays is known as a Stochastic Petri Net [FLO 85]. If we allow immediate firing transitions (for synchronisation modelling), the model obtained is the Generalised Stochastic Petri Net. In both cases, the successive marking of the net can be represented by a Markov Chain and therefore, dependability will be evaluated analytically. Many dependability studies on computer systems use this method [FOT 97].

More generally, it may be useful to model state changes of a device that do not represent fault or repair process, but a change related to the regular behaviour of the system. In this case, the delay of the state change is modelled by the designers with a distribution function on a time interval [ERE 96]. Dependability results are then generally obtained by Monte-Carlo simulation: many histories are simulated during the mission time and the average number of histories that reached feared event is computed.

The delay of a state change may also depend on the physical evolution of a continuous process. Inversely, the configuration of the system influences the evolution of the continuous process. This is typically the case in mechatronic systems where the control system is more particularly devoted to constrain some process variable within specified limits. As a consequence of an initiating event, some process variable might cross these limits, and the control system modifies the system configuration to influence the evolution of the process and bring back the system between its regular limits.

This hybrid point of view is essential to evaluate dependability of mechatronic systems. Indeed, both continuous and discrete parts dynamics influence the dependability of the mechatronic system. Reconfigurations will succeed only if it take place during a « grace period » which goes from the date when the control boundary is exceeded to the date when the feared event occurs. The duration of the « grace period » depends on the dynamic of the operative part and the duration of a reconfiguration depends on the control system and actuators dynamics [MAR 96].

Today, tools for modelling and simulation of hybrid systems exist. The control part is modelled by means of Petri Nets or State Charts. The continuous part is generally modelled by differential algebraic equations. But it is yet difficult to achieve numerical integration for Monte-Carlo simulation in a reasonable time.

A way of solving the problem is to derive an abstract model of the operative part. Indeed, it is often possible to transform differential algebraic equations into explicit and purely algebraic ones. By means of Coloured Petri Nets (CP-nets), the operative part will be modelled in this way. The behaviour of all parts of the system can be captured in a CP-net.

For all these reasons, Coloured Petri Nets [JEN 92, JEN 94] were chosen to model our system for simulation purposes. We use the DesignCPN tool [JEN 97].

## 2. CASE STUDY AND MODELLING

### 2.1 Case study

We study a simple mechatronic system (Figure 2), derived from a more complex system, whose purpose is to maintain a level of pressure (P) in the range [Pmin,Pmax]. The functional constraints are given here after:

- If  $P > P_{max}$  then electrovalve is closed,
- If  $P < P_{min}$  then electrovalve is opened,
- If  $P > P_{alarm\_max}$  or  $P < P_{alarm\_min}$  then the system fails

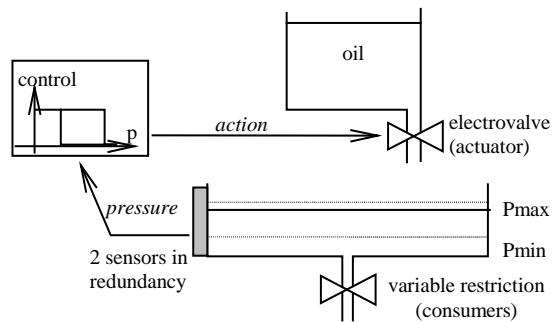


Figure 2 : Principle of the studied system

### 2.2. Functional and dysfunctional model

The only fault considered here is the leak of the tank. The system can be then modelled by algebraic relations as follows (P is the tank pressure, V the volume,  $Q_{in}$  the input flow,  $Q_{out}$  the output flow,  $Q_{consumers}$  the consumption flow,  $Q_{leak}$  the leak flow and  $Q_{pump}$  the pumpflow):

$$(r1): P = f(V), f \text{ reversible,}$$

$$(r2): V = (Q_{in} - Q_{out}) * (t - t_0) + V_0 \text{ with } Q_{out} = Q_{consumers} + Q_{leak}, Q_{consumers}(t) \text{ and } Q_{leak}(t) \text{ can be whatever step functions,}$$

$$(r3): \text{if the electrovalve is opened then } Q_{in} = Q_{pump} \text{ else } Q_{in} = 0.$$

Relation (r1) and (r2) are explicit algebraic equations, which allows to compute the date of a threshold overshoot by P ( $P = P_{threshold}$ ), given an initial state ( $P = P_0, V = V_0$  at time  $t = t_0$ ):

$$t_{threshold} = t_0 + [f^{-1}(P_{threshold}) - V_0] / (Q_{in} - Q_{out})$$

### 2.3. CP-nets

We suggest to represent our system by three successive CP-nets ranging from the more easy to read to the more efficient in terms of simulation time.

The so called « specification » model describes at each sampling date the interactions between operative part and control. This representation is close to the way of thinking of the designers. The principle of the CP-net is the following (Fig. 3): for each device, one place contains a token describing the state of the system and the associated

starting date. In our example, such a state is described by the pressure and volume in the tank, the phase of the system (position of the electrovalve), the level of consumption and the kind of leak. Each time a discrete event occurs like a change of consumption, a change of electrovalve position or the occurrence of a leak (transitions *Change Consumption*, *Change Actuator* or *Leak* are respectively fired) the state is updated and a time stamp is associated to the token indicating the date when the next feared event will appear (transition *Failure* fired). The present state is calculated knowing the previous state and associated starting date, according to relations (r1) and (r2). At each sampling date, the control system read the pressure in the tank (transition *Read Pressure*) and update the command (place *pos req*).

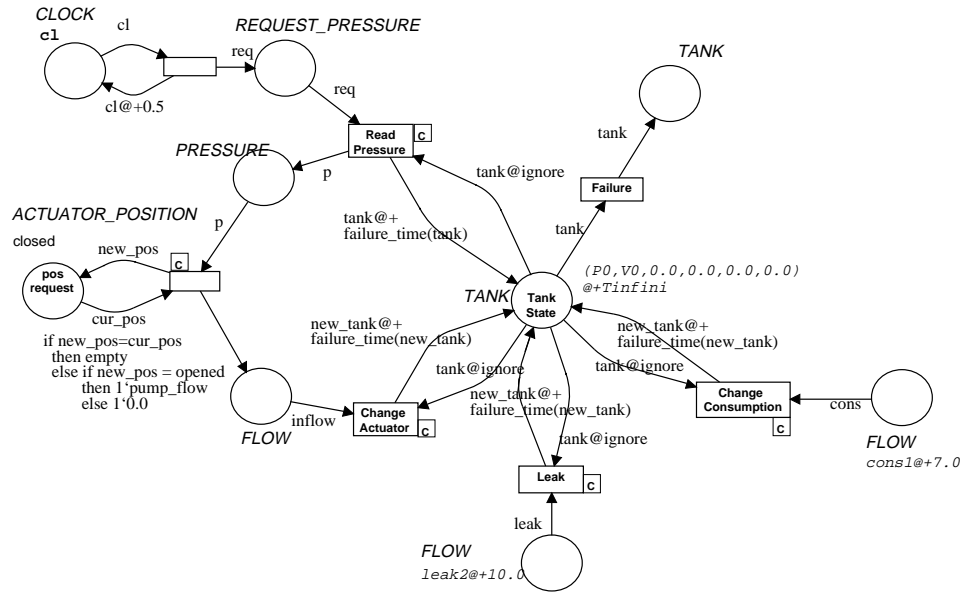


Figure 3 : The « specification » CP-net

We can remark that calculating the new command at each sample date is useless. Indeed, the command changes only once given pressure limits are overshoot. Three pressure intervals are sufficient to describe the command effect. We must add two prohibited intervals which define the feared events (Figure 4).

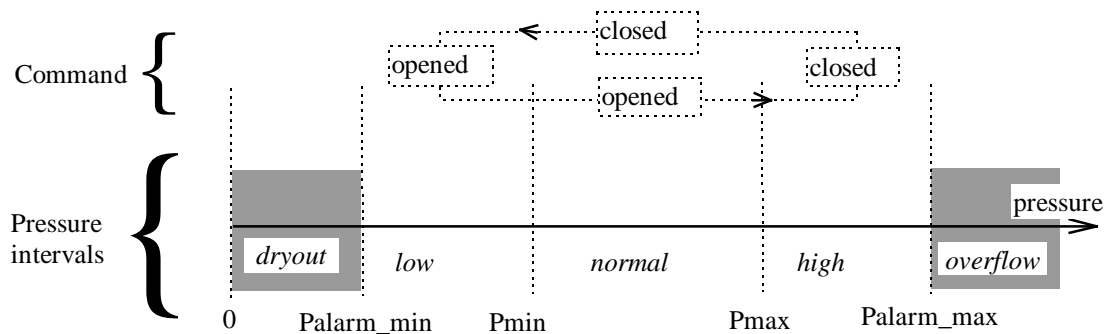


Figure 4 : Pressure intervals and the related command

By using these predefined intervals, it's possible to build an abstract model which represents the same behaviour as modelled by the « specification » model (Figure 5).

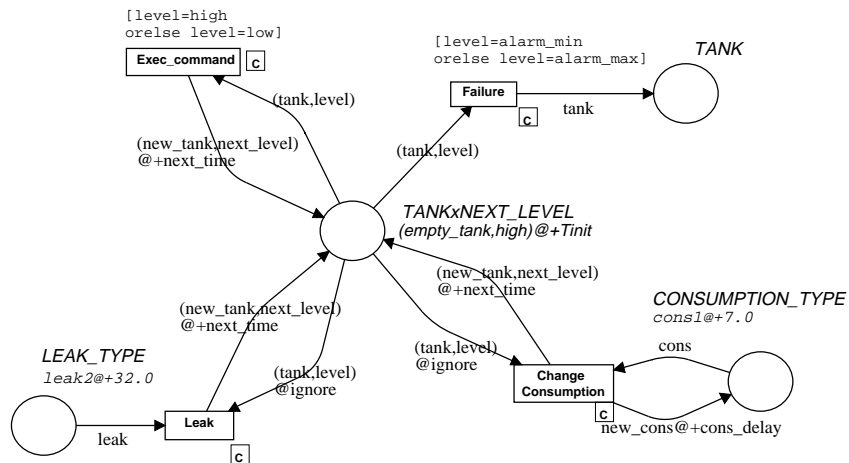


Figure 5 : The abstract CP-net

Each time an external event occurs (leak or change of consumption), the new state (pressure and volume in the tank, pressure interval, position of the electrovalve, level of consumption and the kind of leak are contained in the *new\_tank* variable on arc inscriptions), the occurrence date and the current pressure interval are updated. Then, the next interval that will trigger a decision (command or failure) and the delay when this interval will be reached are calculated (*next\_level* and *next\_time* variables). A « jump » to the decision level can be realised, the state is then updated. According to the level reached (guards on the transitions *Exec\_command* and *Failure* decide which transition is enabled), the command is calculated or a failure is detected. Moreover, as far as the command is concerned, the next decision level and delay when it will be reached is calculated. Notice that a threshold overshoot is detected by the control system after a delay due to the sampling. This effect can be taken into account by adding to the ideal date of a command switch a small  $\Delta t$  corresponding to this delay. Behaviours of both models are then strictly the same (Figure 6).

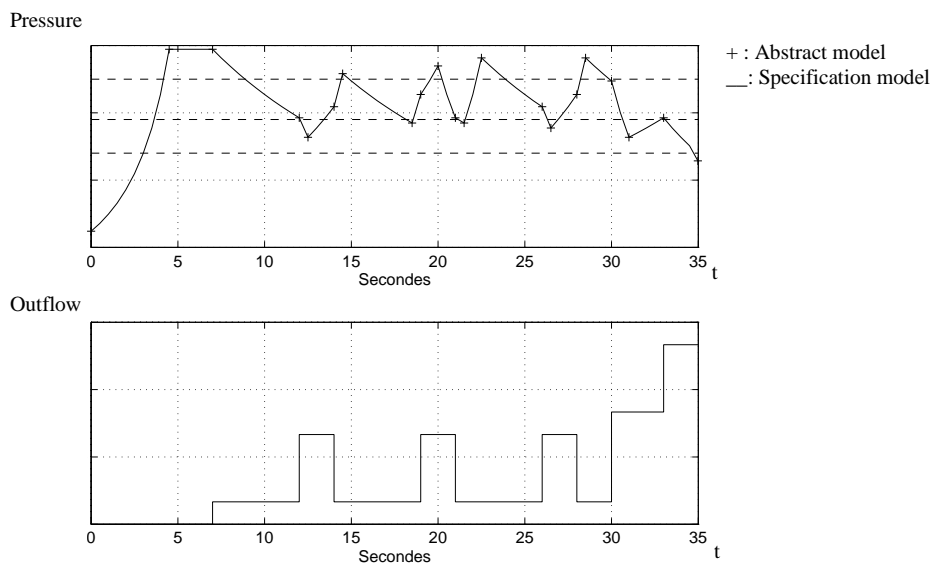


Figure 6 : Comparison of behaviours simulated by the specification and abstract models with a 0.5 seconds sampling period

At last, an optimised and equivalent model is deduced from the previous one, so that simulation can be done without the simulator of the tool DesignCPN. This model is written in ML code (the functional language used by

DesignCPN) and shares the data structures and functions used by the abstract model. As shown on Table 1, these successive models allow to improve simulation times.

	Specification model	Abstract model	ML function
Simulation time in seconds	745	75	1.5
Acceleration	1	10	500

Table 1 : Computation times for a 10 hours mission run and a 0.5 seconds sampling period

### 3. DEPENDABILITY EVALUATION

The system described till now fails only if a leak occurs. Since this fault leads surely to the dryout feared event, it is not necessary to build such a model to evaluate probability to get into the dryout state : we only need to have the leak probability. But suppose that one failed component can be repaired in a very variable time (several reparations may be needed) and that the time available for the reparation depends on the continuous process. We need then to model the system from this hybrid point of view.

This is actually the case we study now. The electrovalve is an actuator which may fail on demand and remain blocked in the current position. Designers experimented that if the electrovalve is blocked in closed position, it may be possible to unblock it by "shaking" it (electric pulse train) during a time called *Tshake*. If the shaking failed, the shaking will be repeated after an idle period which lasts *Twait*. This strategy is also useful when the pump has failed. Indeed, it also is possible to prime it again if the electrovalve move quickly between its extreme positions, as it is the case when it is shaken.

This new control policy could be chosen each time the pressure get under a pressure limit called *Pshake* (due to the closed blocked electrovalve or a pump failure), and abandoned as soon as the pressure grows again.

#### 3.1. Quantitative modelling

We modelled the electrovalve faults, and the corresponding control policy as suggested before. The abstract model is given in Figure 7.

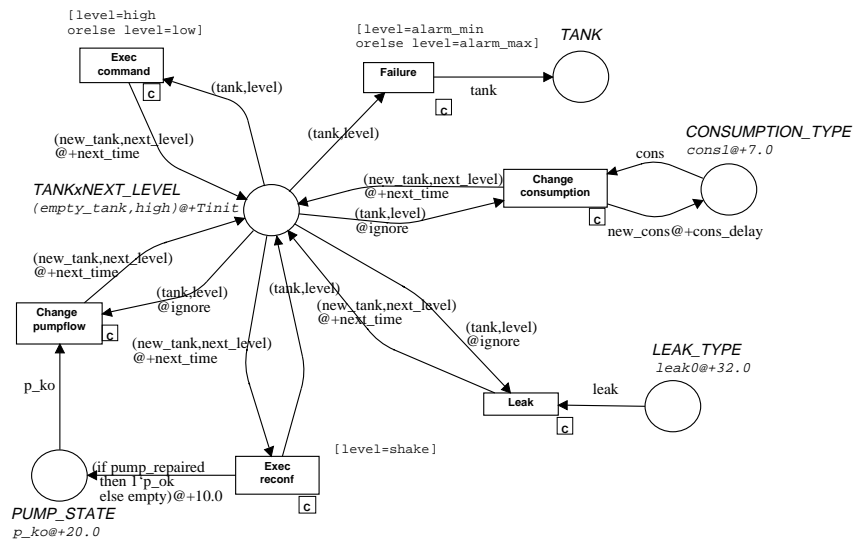


Figure 7 : Abstract model with possible repair of the pump and electrovalve

Monte-Carlo simulations showed that for a given set of parameters (mission time, sampling period, fault rates, *Pshake*, *Tshake*, *Twait* and consumption profile), the probability occurrence of the "dryout" event at the end of the mission time could be divided by ten with a 50 % of reconfiguration success for the electrovalve, and 50% of reconfiguration success for the pump (Figure 8). We could so observe and show the interest of this strategy. A simulation of 1000 histories each corresponding to 1000 hours of real time behaviour takes 40 hours.

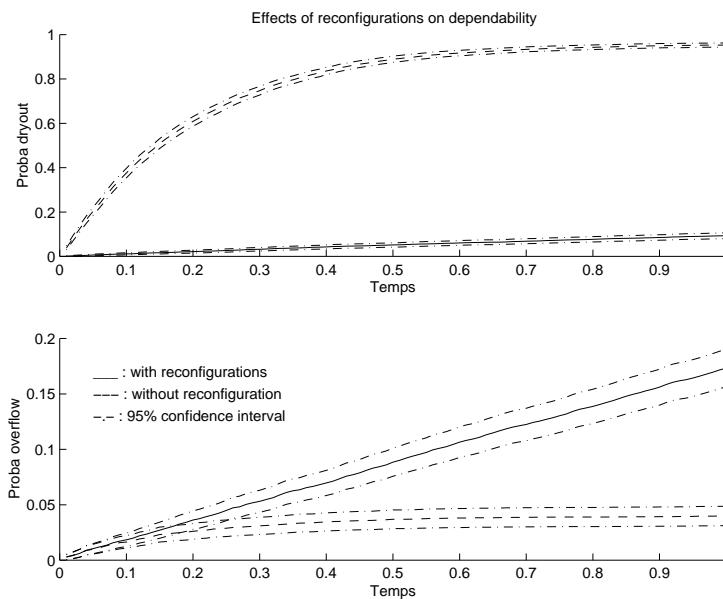


Figure 8 : Effects of reconfigurations on dependability

We can notice that one side effect of the reconfiguration used is that the probability to get the « overflow » event increase with time more quickly than without the reconfiguration. This is due to the fact that failure rates are very high : less « dryout » events let more chance to get « overflow » events.

### 3.2. Qualitative modelling

The Occurrence Graph is generally used to verify dynamic properties like boundedness or liveness properties, and more generally to validate the model. It contains all the possible states the system can reach and how they are reached (an arc represents a transition firing and so an event, a place represents a PN marking and so a state of the system).

But the Occurrence Graph cannot be exhaustive, and so useful for a model validation, if tokens can take an infinite number of possible values. That is the case in our models where time is explicitly represented by a continuous variable. Indeed faults may happen at any time and so an infinite number (non countable) of histories can occur.

A solution is to consider the associate qualitative model where time is not handled any more, but only the order of the events sequences is considered. In the previous quantitative model, the domains of the continuous process variables is naturally divided into discrete levels. The evolution of the system is determined by the actual discrete state and the occupancy duration of this state. To get the associate qualitative model, we only need to eliminate explicit time references. Time will be modelled then by the order when events occur. The Occurrence Graph of the associate qualitative model can be so completely build. The qualitative model of the studied system is given in Figure 9 (the Occurrence Graph contains 21 nodes and 34 arcs).

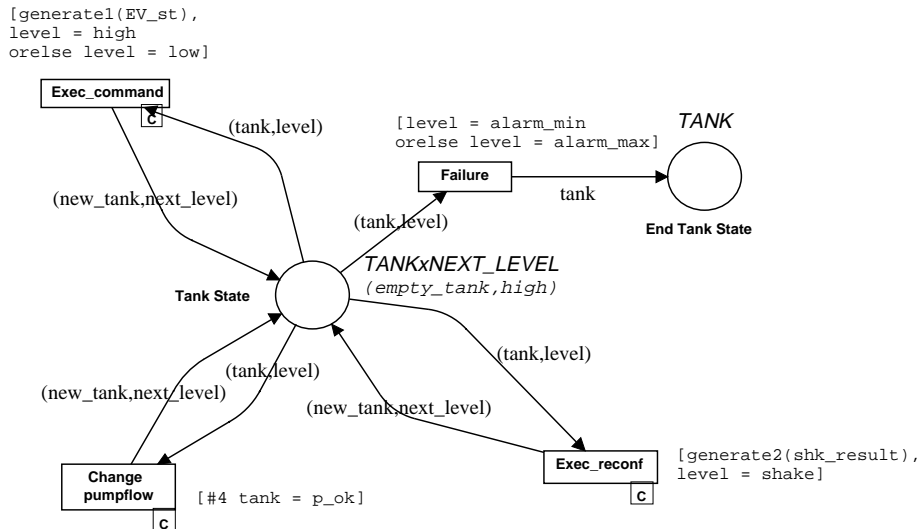


Figure 9 : Qualitative model derived from the abstract model of Figure 7

To validate the model, we can prove, for example, that dead markings represent no other states than faulty states (no dead markings due to a dead lock). We can also verify that some well known scenarios happen as expected by building it directly thanks to the Occurrence Graph tool or the simulation tool.

We can also explore systematically all the possible scenarios (and possibly find some unexpected ones). This analysis is based on the strongly connected component (Scc) of the occurrence graph.

Indeed, the notion of Scc has a useful interpretation from the dependability point of view. Any state of a Scc can be reached from any other state of the Scc (Figure 10). Two non exclusive cases are possible in the case of hybrid systems :

- the Scc contains a set of states covered cyclically during a possible infinite time which corresponds to the supply of a service in a nominal or degraded operating mode,
- the Scc represents a transitory evolution which ends when conditions of a feared event are detected (when a process variable crosses a given threshold, the system reaches an absorbing state ).

Any arc which goes out of a Scc means that the associated event prevents the system from going back to the

previous operating mode or transitory state. These events are called critical events (thick arrows on Figure 10). There are two kinds of critical events :

- non repairable faults,
- detection of a feared event occurrence.

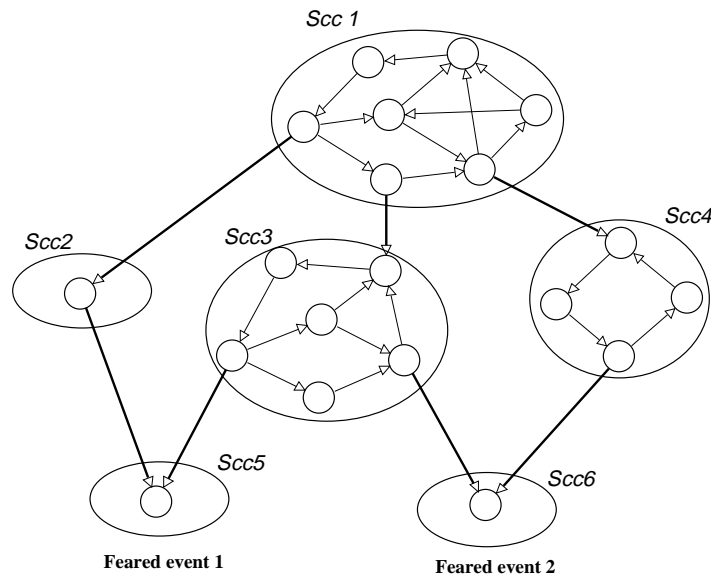


Figure 10 : Example of Strongly Connected Component Graph

The analysis is done in three steps and consists on :

- defining a partition of dead markings. Each of the disjoint sets corresponds to one expected feared event,
- interpreting all the non trivial Sccs and finding the functioning mode it represents,
- finding and interpreting all the possible paths (sequence of events - one event is characterized by a binding element) from each Scc to the next Sccs or feared events. These paths define trees of bindings and one branch describes one of the sequences of events that occur when the system leaves the considered functioning mode. Many branches of these trees can be merged if we group together bindings that represent the same events.

This work of interpretation involves much of the knowledge the designer have on his system and cannot be done automatically. It could be difficult and require a long time for complex systems.

Note that the dead markings describe generally the set of faults which lead to the considered feared event. But it gives no information on the order in which these faults occurred, which may be non trivial for complex systems. Indeed, some sequence of faults may lead to a feared event whereas the same set of faults in a different order may not. Moreover, a fault may trigger a feared event only in presence of a particular solicitation of the system. In this case, the dead markings give no information about the sequence of solicitations (combined with faults) that lead to the feared events. The analysis of the paths going out of the Sccs is so generally essential.

For our example, we can prove that only one Scc (related to a total of 15 Scc nodes) contains a cycling set of states corresponding to the support of the acceptable level of pressure. We can prove that only two kinds of events trigger the exit of this Scc and directly lead to feared events :

- a reconfiguration failed to repair the pump or the closed blocked electrovalve,
- the electrovalve remained opened blocked when asked to close.

#### 4. CONCLUSION AND PROSPECTS

CP-nets is well adapted to describe an hybrid system model, with the assumption that the operative part can be modelled by explicit algebraic equations. Moreover, an ML code for Monte-Carlo simulations can be deduced and validated through an abstract CP-net. We first got dependability evaluation using this method on a simple mechatronic system.

To complement this quantitative dependability evaluation approach, an important issue is to determine all the possible scenarios, particularly rare scenarios which cannot be easily shown by Monte-Carlo simulation. We showed how useful is the DesignCPN occurrence graph analysis tool in some simple case.

The next step will be to use these approaches on a more consequent system.

#### BIBLIOGRAPHY

- [ERE 97] Jean-François Ereau et Malecka Saleman: « Modelling and Simulation of a Satellite Constellation based on Petri Nets », Annual Reliability and Maintainability Symposium, Proceedings 1996.
- [FOT 97] Nicolae Fota: « Spécification et Construction Incrementale de Modèles de Sûreté de Fonctionnement - Application au CAUTRA », thèse présentée au LAAS, 1997.
- [FLO 85] G. Florin et S. Natkin: « Les réseaux de Petri stochastiques », Techniques et Sciences Informatiques, vol. 4, n°1, 1985.
- [GUY 94] Jacques Guyot: « Mechatronic components design in the automotive industry », Proceedings of the 2nd Japan-France congress on Mechatronics, Japan, 1994.
- [HEN 96] Valéry Hénault: « Méthodologie de développement des systèmes électroniques embarqués automobiles, matériels et logiciels, sûrs de fonctionnement », thèse présentée à l'IRESTE, septembre 1996.
- [JEN 92] Jensen, K. (1992) Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use. Vol. 1, Basic Concepts. EATCS Monographs on Theoretical Computer Science, Springer-Verlag.
- [JEN 94] Jensen, K. (1994) Coloured Petri Nets - Basic Concepts, Analysis Methods and Practical Use. Vol. 2, Analysis Methods. Monographs in Theoretical Computer Science. Springer-Verlag.
- [JEN 97] Jensen, K.; Christensen, S.; Huber, P.; Holla, M. (1997) Design/CPN Reference Manual. Computer Science Department, University of Aarhus, Denmark. On-line [http //www.daimi.aau.dk/designCPN/](http://www.daimi.aau.dk/designCPN/).
- [LER 92] Alain Leroy et Jean Pierre Signoret: « Le risque technologique », Collection « Que sais-je ? », 1992.
- [MAR 96] M. Marseguerra & E. Zio: Monte Carlo approach to PSA for dynamic process systems, Reliability Engineering & System Safty, vol. 52, 1996
- [PAG 80] A. Pagès et M.Gondran: « Fiabilité des systèmes », collection de la Direction des Etudes et Recherche d'Electricité de France, 1980.