A formal definition of Minimal Cut Sequences for dynamic, repairable and reconfigurable systems

Pierre-Yves Piriou, Jean-Marc Faure, Jean-Jacques Lesage

A formal definition of Minimal Cut Sequences for dynamic, repairable and reconfigurable systems

P.Y Piriou[1], J.M. Faure[2] & J.J. Lesage[3]

1 - Electricité de France, R&D, 78400 Chatou, France
2 - LURPA, ENS Cachan, Univ. Paris-Sud, Supmeca, Univ. Paris-Saclay, 94235 Cachan, France
3 - LURPA, ENS Cachan, Univ. Paris-Sud, Univ. Paris-Saclay, 94235 Cachan, France

ABSTRACT: MCS (Minimal Cut Sequences) computation is the main objective of qualitative safety analysis of dynamic systems. This paper shows first that the existing definitions of MCS are not suitable when systems are repairable and reconfigurable, then proposes a new definition for this class of systems. This proposal is illustrated on a case study from power industry. Comparison of the obtained MCS to those which are yielded by algorithms based on previous definitions permits to highlight the relevance of the approach.

1 INTRODUCTION

Modern systems are expected to be more and more flexible and dependable. To deal with these needs, system designers have defined reconfiguration strategies, i.e. switching mechanisms that change on-line the system structure and/or behavior. Such reconfigurations can be motivated by functional requirements (e.g. to change the phase of the mission), fault tolerance objectives (management of redundant resources), predictive maintenance policies, production needs etc... The complexity and diversity of reconfiguration strategies are even more important when the system is repairable. These reconfiguration strategies have a significant impact on system safety. In particular, we have shown in [Piriou2014] that the failure of the control part that manages the reconfiguration strategies may have a huge impact on the failure of the whole system.

Qualitative safety analysis aim to link the failure of the system to the failure of its components, what provides important expert knowledge in the design process. For dynamic systems, this knowledge has been formalized through the notion of *cut sequence (CS)*: a scenario (sequence of event occurrences) that leads the system from its initial state to a failure state (without passing through another failure state). For dynamic systems, since the set of CS is very huge, Minimal Cut Sequences (MCS - that can be informally defined as the subset of CS sufficient to the knowledge of all CS) have been defined.

Several definitions of MCS coexist in the literature ([Rauzy2011], [Walker2007], [Tang2004], [Chaux2013]). Only the two latter ones will be considered in this paper because the other ones do not consider repairable systems. It will be shown first, by using counter-examples, that the definitions of MCS which are proposed in these references are not suitable for reconfigurable systems. A new formal definition will be afterwards proposed, based on the postulate that a dysfunctional sequence is characterized by the order of the events occurrences it includes and by the set of faulty components at the end of the sequence.

In order to illustrate the benefit of this new definition for repairable and reconfigurable systems, the paper proposes next an algorithm for computing MCS from dysfunctional models built by using Generalized Boolean logic Driven Markov Processes (GBDMP). A case study is presented for illustrating both the GBDMP modeling power and the MCS computing principle. The set of MCS obtained is afterwards compared to the one obtained by using the previous definitions of MCS and the differences are analyzed in terms of implication for safety.

The outline of the paper is the following. In first section, a selection of existing definitions of MCS are recalled. Section 2 proposes a new definition allowing to deal with dynamic repairable and reconfigurable systems. Section 3 shows how the MCS set can be computed from a GBDMP model. A case study is addressed in section 4 to illustrate the approach. Finally, concluding remarks and perspectives are drawn up in section 5.

2 BACKGROUND

[Chaux13] defines the MCS set as the minimal set of sequences of minimal length that are necessary and sufficient to represent the whole set of cut sequences. In this work, the main problem consists in

formally defining the order relation "*to represent*" between two cut sequences.

The usual notations and vocabulary of the language and automata theory ([Meduna2012]) are used in what follows.

2.1 *A language-theory-based problem statement*

Let us note:

- Σ the alphabet of events that are to be considered for safety analysis (failures, repairs, etc.).
- $\mathcal{L}_E \subseteq \Sigma^*$ the evolution language of the system, i.e. the set of all scenarios built on Σ that may happen within the system.
- $\mathcal{L}_F \subseteq \mathcal{L}_E$ the failure language of the system i.e. the set of all evolution scenarios that lead the system from its initial state to a failure one.
- $\mathcal{L}_{CS} \subseteq \mathcal{L}_F$ the set of cut sequences:
  $\mathcal{L}_{CS} = \{ \sigma \in \mathcal{L}_F \mid \forall \sigma' \in \text{Pref}(\sigma) \wedge \sigma' \neq \sigma, \sigma' \notin \mathcal{L}_F \}$.
- $\mathcal{L}_{MCS}(\mathcal{R}) \subseteq \mathcal{L}_{CS}$ the set of Minimal Cut Sequences according to the order relation $\mathcal{R}$:
  $\mathcal{L}_{MCS}(\mathcal{R}) = \{ \sigma \in \mathcal{L}_{CS} \mid \forall \sigma' \in \mathcal{L}_{CS},\ \sigma' \mathcal{R} \sigma \Rightarrow \sigma' = \sigma \}$

In this definition, the problem is to define the order relation $\mathcal{R}$ that relevantly translates the informal relation "*to represent*".

To illustrate these definitions, let us consider a toy example: two components A and B in standby redundancy driven by a third control component C. When C fails, components A and B remain stuck in their current configuration until C be repaired. The system is faulty when the currently active component (A or B) is faulty. The automaton $\mathcal{A}$ that models the dysfunctional behavior of this system is depicted on Figure 1[1]. For this automaton, we have: $\Sigma = \{f_A, r_A, f_B, r_B, f_C, r_C\}$, $\mathcal{L}_E = \mathcal{L}(\mathcal{A})$, $\mathcal{L}_F = \mathcal{L}_m(\mathcal{A})$ and $\mathcal{L}_{CS} = \mathcal{L}_m(\mathcal{A}')$, where $\mathcal{A}'$ is the automaton $\mathcal{A}$ for which all transitions leaving a marked state have been removed.

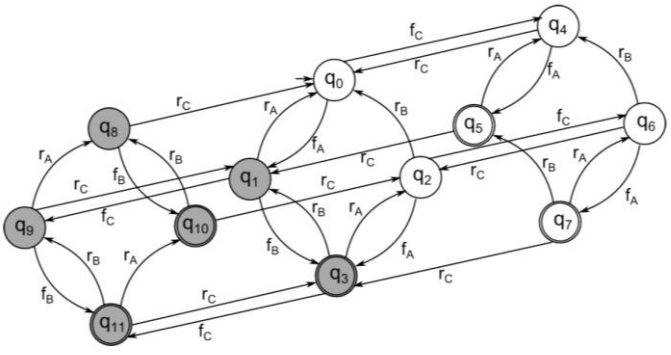


Figure 1. Automaton modeling two components A and B in standby redundancy driven by a third control component C.

For any evolution sequence $\sigma \in \mathcal{L}_E$, let us denote its *covering cut* $[\sigma]$ the set of components that are faulty at the end of σ. For the considered example, $[f_A f_B] = \{A,B\}$, $[f_A f_C r_A f_B] = \{B,C\}$…

2.2 *First basic definition of MCS*

The first proposition for $\mathcal{R}$ is the *sequence inclusion*, denoted "$\subseteq$" (see [Tang2004] and [Chaux2012]): a sequence σ is included into another sequence σ' if and only if all events of σ are in σ' in the same order. This definition is convenient only for non-repairable systems. Indeed, for the example given at Figure 1, we have: $\mathcal{L}_{MCS}(\subseteq) = \{f_A f_B, f_C f_A\}$. In particular, $f_A f_C r_A f_B \notin \mathcal{L}_{MCS}(\subseteq)$ because $f_A f_B \subseteq f_A f_C r_A f_B$. Nevertheless, the longer sequence expresses supplementary information on the system dysfunctional behavior: the effect of the failure of C (the resumption of A after its repair cannot be processed). Then $f_A f_B$ does not represents $f_A f_C r_A f_B$ according to the safety point of view, and $\subseteq$ is therefore not a suitable order relation for repairable systems.

2.3 *A definition of MCS based on coherence rules*

A promising definition of MCS for dynamic and repairable systems has been proposed in [Chaux2013]. It is based on a definition of the coherence: a system is said *coherent* if and only if, starting from any failure sequence ($\sigma \in \mathcal{L}_F$) by adding events according to the two following applications a new failure sequence is obtained:

1. insertion of a single failure event.
2. ordered distribution of a set of events that let unchanged the covering cut.

Let us call respectively $f_1(\sigma) \subseteq \mathcal{L}_E$ and $f_2(\sigma) \subseteq \mathcal{L}_E$ the sets of all possible sequences that can be obtained by adding events $\in \Sigma$ to σ by using applications 1 and 2 mentioned above. f_1 and f_2 can easily be extended to languages ($\forall \mathcal{L} \subseteq \mathcal{L}_E$, $f_1(\mathcal{L}) = \bigcup_{\sigma \in \mathcal{L}} f_1(\sigma)$ and $f_2(\mathcal{L}) = \bigcup_{\sigma \in \mathcal{L}} f_2(\sigma)$). A coherent system can then be defined as a system verifying the following property:

$$\forall \mathcal{L} \subseteq \mathcal{L}_F,\ \forall n \in \mathbb{N},\ f_2(f_1^n(\mathcal{L})) \subseteq \mathcal{L}_F \quad (1)$$

According to [Chaux2013], a sequence σ "represents" another sequence σ' if and only if σ' can be coherently built from σ (i.e. by using f_1 and f_2). Hence a new proposition of $\mathcal{R}$ definition arises: the *coherence relation* $\vDash$.

$$\forall (\sigma, \sigma') \in \mathcal{L}_E^2,\quad \sigma \vDash \sigma' \Leftrightarrow \exists n \in \mathbb{N} \mid \sigma' \in f_2(f_1^n(\sigma)) \quad (2)$$

For the example depicted in Figure 1, with definition (2), $\mathcal{L}_{MCS}(\vDash) = \{f_A f_B, f_C f_A, f_A f_C r_A f_B\}$ what is satisfactory. Nevertheless the next section shows that this definition is not suitable for reconfigurable systems.

[1] f_X: failure of X ; r_X: repair of X ; white states: A is active and B is inactive ; grey states: A is inactive and B is active. The marked states are double-circled.

3 A NEW DEFINITION OF MCS

3.1 *Limitation of the coherence- based definition of MCS*

The dysfunctional behavior of reconfigurable systems cannot be always described with only failure and repair events. For example, Figure 2 shows the behavior of a system including two components (A and B) that perform a mission in two phases. During the first phase A and B are in standby redundancy, whereas during the second one both are required. On Figure 2, φ represents the event of phase switching.

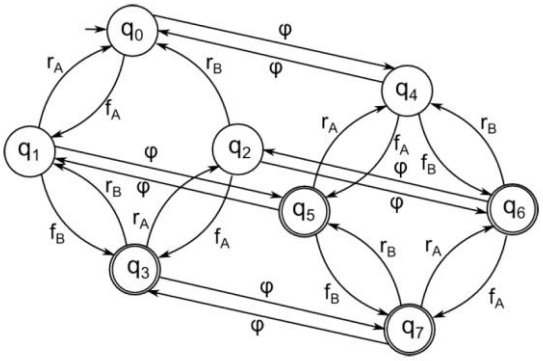


Figure 2. Automaton modeling two components A and B that perform a phased mission.

This system is not coherent according to (1). Indeed, the inserting of event $\varphi \in \Sigma$ at the end of the failure sequence φf_A does not result in a failure sequence although the covering cut remains unchanged. Therefore the notion of coherence cannot be directly extended to all reconfigurable systems.

3.2 *A new minimality criterion for CS*

For dynamic systems the relative order of event occurrences matters to determine the system failure. Therefore the set of MCS should be defined using a relation that preserves this order (like the *sequence inclusion*). Nevertheless, as for static systems, the dysfunctional behavior of a dynamic system is linked to the failure of its components. The set of MCS should consequently be defined using a relation that preserves the set of faulty components (like the *set inclusion*).

This observation allows us to postulate that a cut sequence is characterized not only by the order of events but also by its covering cut. Let us remark that this postulate complies with the nature of applications f_1 and f_2 defined in subsection 1.3.

Based on this postulate we can propose a new order relation to define the MCS set: the *cut sequence inclusion* $\Subset$.

$$\forall(\sigma,\sigma') \in \mathcal{L}_E{}^2, \quad \sigma \Subset \sigma' \Leftrightarrow (\sigma \subseteq \sigma') \wedge ([\sigma] \subseteq [\sigma']) \qquad (3)$$

This relation is less restrictive than the *coherence relation*. Indeed, we can easily check (4) but the inverse is not true.

$$\forall(\sigma,\sigma') \in \mathcal{L}_E{}^2, \quad \sigma \vDash \sigma' \Rightarrow \sigma \Subset \sigma' \qquad (4)$$

In particular, the relation $\Subset$ allows to determine satisfying sets of MCS for the first example (cf. Figure 1): $\mathcal{L}_{MCS}(\Subset) = \mathcal{L}_{MCS}(\vDash)$; and for the second (cf. Figure 2): $\mathcal{L}_{MCS}(\Subset) = \{f_A f_B, f_A \varphi, \varphi f_A, \varphi f_B\}$. The next section shows how this new definition can be used to compute the set of MCS from a GBDMP model.

4 COMPUTING MCS FROM A GBDMP MODEL

4.1 *Recall on GBDMP formalism*

Generalized Boolean logic Driven Markov Processes (GBDMP) is an extension for reconfigurable systems of the BDMP formalism defined in [Bouissou03] for safety analysis of dynamic and repairable systems. Basically, a GBDMP integrates in the same model a representation of the system structure (Fault Tree), the dysfunctional behaviors of the components of the system (Switched Markov Processes) and the reconfiguration mechanisms (Moore Machines).

The bases of GBDMP syntax and semantics are now briefly given and exemplified. They have been formally and broadly presented in [Piriou2016].

Definition 1. A Generalized Boolean logic Driven Markov Process is a 6-tuple <*V, E, κ, υ, str, smp*> where:

- $V = N \cup S = G \cup L \cup S$ *is a set of vertices partitioned into the nodes N (i.e. the gates G and the leaves L) and the switches S;*
- $E = E_F \cup E_S$ *is a set of oriented edges, such that* $E_F \subseteq G \times N$ *and* $E_S \subseteq (N \times S) \cup (S \times N)$*;*
- $\kappa: G \rightarrow \mathbb{N}^*$ *is a function that determines the gates kind. This function is the same as the one used in BDMP [2];*
- $\upsilon : E \rightarrow \mathbb{N}$ *is a function that associates an integer label to each edge;*
- $str: S \rightarrow \mathbb{M}$ *is a function that associates a Moore machine (which represents a reconfiguration strategy) to each switch.* $\mathbb{M}$ *designates the set of Moore machines;*
- $smp: C \rightarrow \mathbb{P}$ *is a function that associates a SMP to each component (a k-SMP for a component with k operation modes).* $\mathbb{P}$ *designates the set of Switched Markov Processes.*

A simple GBDMP is shown at Figure 3. The structure of the system is represented by a fault tree (part a of Figure 3). It is composed of 3 gates (G1 is an AND gate, G2 and G3 are OR gates), 3 basic components (leaves C1, C2, C3 and C4) and a switch (S1 depicted with a dashed rectangle). The

solid (resp. dashed) arrows are the edges of E_F (resp. E_S), which connect the gates to the nodes (resp. the switches to the nodes and the nodes to the switches). The dysfunctional behavior of the leaves C1, C2 and C3 is depicted by the SMP "Pu" at part b of Figure 3. The component C4 is in charge of the control of the switch, its dysfunctional behavior is depicted by the SMP "Co" at part b of Figure 3. The reconfiguration strategy implemented in switch S1 is modelled by the Moore machine at part c of Figure 3. The label, given by function υ, of an edge of E_F and E_S permits to associate respectively a branch to an operation mode of a leaf and a number of input or output of the Moore machine to a node.

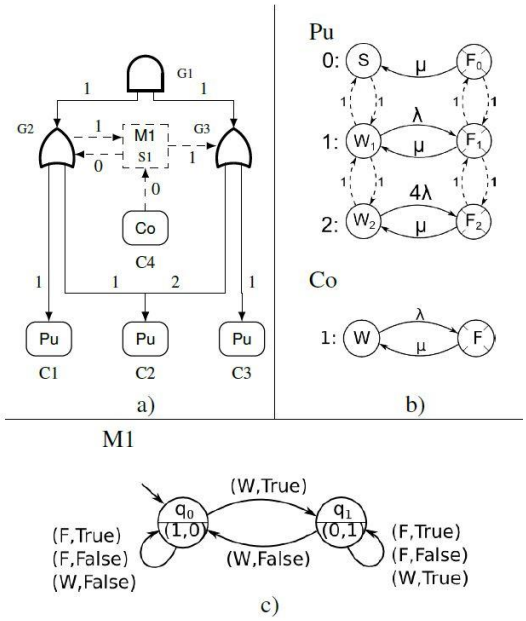


Figure 3: Example of GBDMP. a) Structure modelling; b) SMP Pu (associated to C1, C2 and C3) and Co (associated to C4); c) Moore machine M1 (associated to S1)

The behavior of a leaf is modeled by a *k*-SMP which is composed of *k* Markov chains. Each Markov chain corresponds to an operation mode and comprises faultless and faulty states; the transitions between these states are stochastic and they model mainly failures and repairs. In the example of Figure 3 b, the 3-SMP associated to the leaves C1, C2 and C3 comprises three Markov chains (one for each line) to represent a component with two working modes and one standby mode; in this model, it is assumed that no failure occurs in the standby mode and that the failure rate in the second working mode is greater than the corresponding rate in the first working mode. $k(k-1)$ probabilistic transfer functions between the chains of a k-SMP must be defined. The value of the transfer function between two states of two different chains (in dashed arrows) is equal to 1 if no failure on-demand is considered (case of Figure 1 b) when the operation mode is changed and belongs to [0,1] otherwise.

The role of a switch is to set/reset the requirement statuses of the nodes that are connected to its outputs according to the values of its inputs and the reconfiguration strategy which is described by the associated Moore machine. In the Moore machine M1 at Figure 3 c, let q_0 be the current state. In this state, C1 is activated in operation mode 1, C2 is activated in operation mode 1 and C3 is deactivated. The transition between state q_0 and state q_1 is fired if the associated condition "(W,*True*)" is true, i.e. if the SMP of C4 (input #0 of S1) is in state W and if the output of Gate G2 (input #1 of S1) is *True*. The firing of this transition implies the change of active state of M1 (which becomes q_1) and the change of outputs values: the requirement of G2 (output #0 of S1) is reset and the requirement of G3 (output #1 of S1) is set. As a consequence, C1 is deactivated, C2 is activated in operation mode 2 and C3 is activated in operation mode 1.

4.2 *Translation of a GBDMP into a finite state automaton.*

For qualitative analysis, assuming that the initial state of a GBDMP is deterministic, its semantics can be seen as a finite deterministic state automaton. This subsection describes how a well-formed[2] BDMP $<V, E, K, \upsilon, str, smp>$ can be translated into an automaton $< \Sigma, Q, q_0, Q_M, \delta >$.

The GBDMP evolution is driven by the occurrence of two types of events. On one hand the *spontaneous* events (solid arrows in Figure 3 b) correspond to the perturbations of the system (mainly failures and repairs of a leaf). On the other hand the *provoked* events (dashed arrows in Figure 3 b) correspond to the reconfigurations of the system (operation mode change of a leaf). Obviously all *spontaneous* events are important for safety, whereas only the *provoked* events that change the failure status of the leaf are. Assuming that components cannot be repaired during an operation mode change, only the *failure provoked* events (e.g. on-demand failures) have to be considered for safety analysis. Hence, alphabet Σ is constituted of the set of all *spontaneous* events (solid arrows of the SMP of all leaves) and *failure provoked* events (dashed arrows from a faultless state to a faulty state in the SMP of every leaf).

The global state of a GBDMP model is fully determined from the local states of every SMP and Moore machine. According to the initial state of Moore machines, it is possible to determine the initial state of SMP, what gives the initial global state q_0 of the GBDMP model. Hence the set of states Q is the set of combinations of these local states reachable from q_0.

The definition of the transition function δ can easily be deduced from the evolution rules of a

[2] In [Piriou2016] 5 syntactic properties guaranteeing that a GBDMP is well-formed are given.

GBDMP, which are formally described in [Piriou2016]. Q can be computed recursively by applying δ from q_0.

Finally, for a given analysis, one must select a gate of the GBDMP whose failure is the undesirable event. Then the set of marked states Q_M is simply the subset of states where the selected gate is faulty.

4.3 *An algorithm to calculate MCS from a GBDMP model.*

Algorithm 1 allows computing the set of MCS on the fly by performing a breadth-first exploration of the state space Q. The breadth-first exploration ensures that for each new cut sequence found, it is possible to determine if it is minimal by comparing it with the already found MCS (lines 14-16). The exploration from a sequence is stopped if one of the two following conditions is met (lines 12-13):

1. The sequence reaches a failure state (because only cut sequences are considered).
2. The sequence reaches a state already visited by itself (because obviously, a sequence with a loop is represented by the same sequence without the loop).

Algorithm 1 – Calculus of MCS from a GBDMP model

Inputs: $< \Sigma, Q, q_0, Q_M, \delta >$ the automaton that translates the behavior of a GBDMP model.
Outputs: $\mathcal{L}_{MCS}(\Subset)$ the set of MCS for the relation $\Subset$.

```
1:  // Initialization
2:  L_MCS := ∅
3:  seqOfLastLength := {ε}
4:  seqOfCurLength := ∅
5:  // Main loop
6:  while seqOfLastLength ≠ ∅ do
7:    for all σ_last ∈ SeqOfLastLength do
8:      q_last := δ(q_0 , σ_last)
9:      for all u ∈ E(q_last) do
10:       q_cur := δ(q_last , u)
11:       σ_cur := σ_last u
12:       if q_cur ∉ Q_M ∧ ∄ σ ∈ Pref(σ_last) |
                                  δ(q_0 , σ) = q_cur then
13:         seqOfCurLength :=
                        seqOfLastLength ∪ { σ_cur }
14:       else if q_cur ∈ Q_M then
15:         if ∄σ_min ∈ L_MCS | σ_min ⋐ σ_cur then
16:           L_MCS := L_MCS ∪ { σ_cur }
17:         end if
18:       end if
19:     end for
20:   end for
21:   seqOfLastLength := seqOfCurLength
22:   seqOfCurLength := ∅
23: end while
```

This algorithm uses the function $\mathcal{E}$: $Q \longrightarrow \mathcal{P}(\Sigma)$ which gives the set of events that may occur in a given state of the GBDMP.

Let us remark that this algorithm can easily be applied with other definitions of MCS, by replacing the relation $\Subset$ at line 15 by another one (assuming that this relation is also based on the growing of sequences).

The complexity of this algorithm is $\mathcal{O}(Card(Q))!^2)$. Despite this complexity, the algorithm is applicable in practice because the set of MCS is computed on the fly by increasing order of length. For qualitative analysis, the longer a MCS is, the less relevant it is. The set of MCS computed by the algorithm in a reasonable time constitutes therefore generally a relevant enough qualitative result (even if incomplete).

5 APPLICATION

5.1 *Controlled coolant feeding system*

In order to illustrate the presented approach, the system depicted in Figure 4 is used. It is a very simplified version of a part of the cooling system of nuclear power plants proposed by the company Electricité de France. Its main function is to feed a downstream system with a cooling fluid by using two groups of pumps. The first group of three pumps is powered by a heavily redundant electric power supply whose components are repairable. The system fails when fluid can no more be provided. Three stages of electric energy supply and two pumping stages can be identified:

- An electric transformer Tr1 (A) connected to the grid, is used to provide low voltage electricity; if it fails, a second transformer Tr2 is available thanks to a standby redundancy.
- A distribution board TEb1 (B) powering a second distribution board TEa1 (D) using one of the two transformers Tr1 or Tr2. A diesel generator Di1 (C) is in standby redundancy with subsystem TEb1.
- The lower level distribution board TEa1 (D) powers the group of extraction pumps Ex1, Ex2 and Ex3 (E), using one of the two possible sources (TEb1 or Di1).
- The fluid is extracted by the group of pumps Ex1, Ex2 and Ex3 (E). Only two pumps are used during operation; if one pump fails the third one is activated (standby redundancy 2 out of 3).
- Pumps Pr1 and Pr2 (F) pressurize the fluid; only one pump is used during operation. If the main pump Pr1 fails, the spare pump Pr2 is activated (standby redundancy 1 out of 2).

The subsystem {B, C, D} is duplicated in order to provide a standby redundancy for powering the ex-

traction pumps. We consider that diesel generators and pumps may fail when they are active or in standby, whereas the other components may only fail when they are active.

Thanks to the multiple redundancies, lots of reconfigurations are possible in this system for maintaining the production, even in case of multiple failures of components.

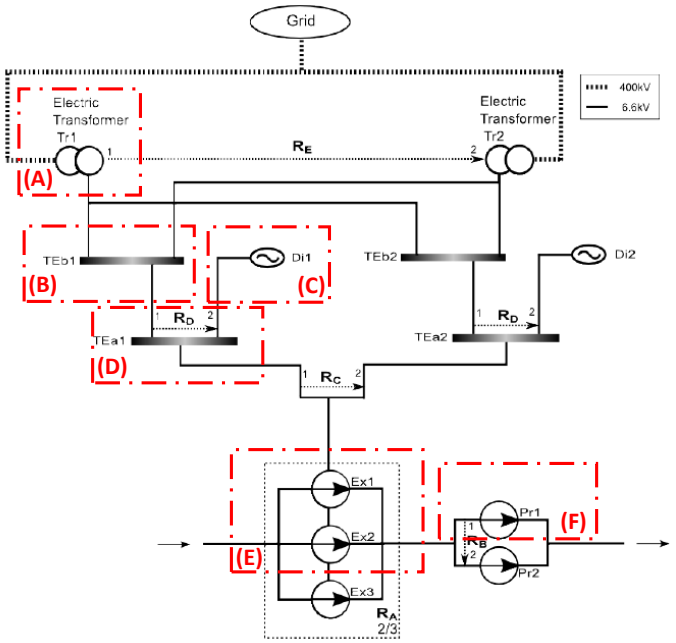


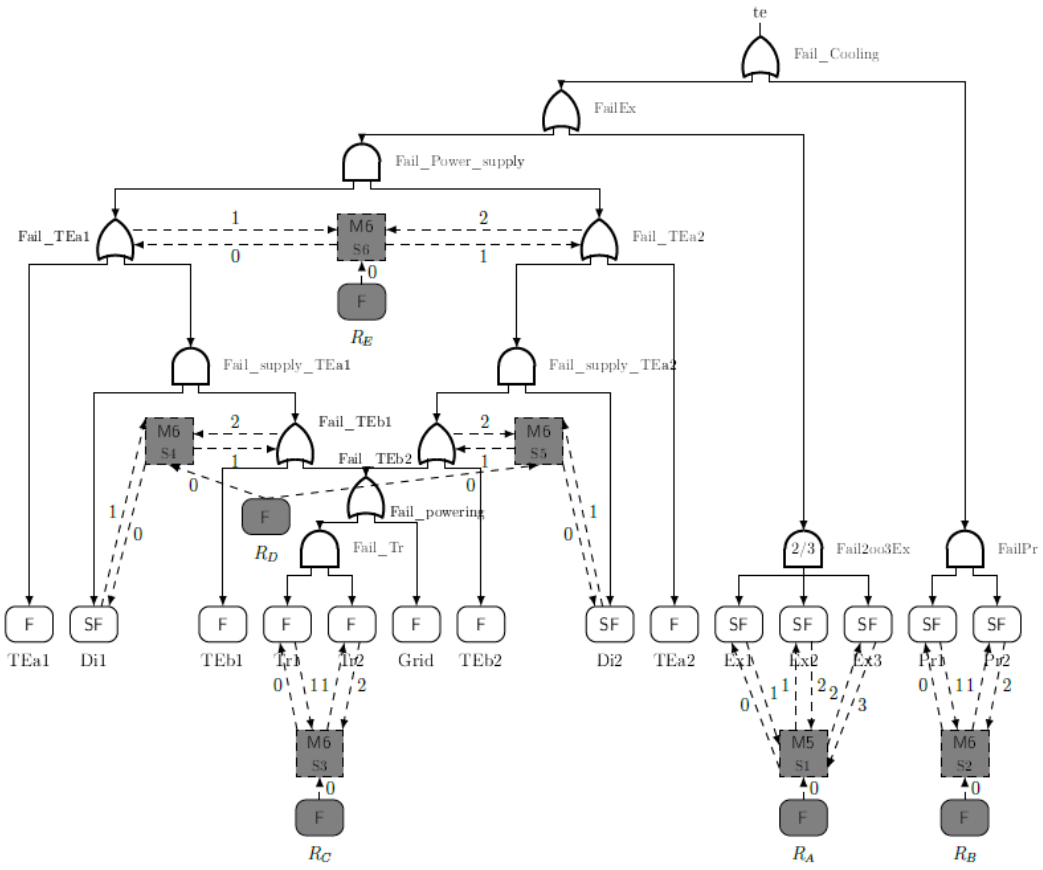
Figure 4. Physical architecture of the coolant feeding system.

5.2 *GBDMP model*

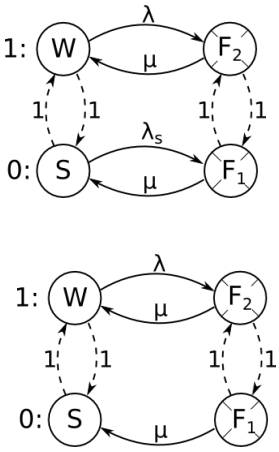
The GBDMP of Figure 5 models the possible dysfunctional behaviors of the coolant feeding system depicted at Figure 4. The structure of the system that makes that certain ordered combinations of failure and repair events of components lead to the global failure *te* (top event) is given by the fault tree Figure 5a.

Each one of the 14 components (including the grid) is associated to a leaf of the GBDMP. Components that may fail in working mode and in standby mode are associated to a leaf of type *SF* (“Standby Failure”); components that may fail only in working mode are associated to a leaf of type *F* (“simple Failure”). The corresponding SMP are given in Figure 5b.

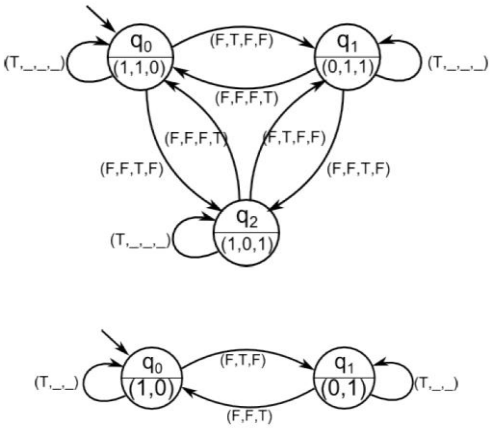
Each time a redundancy between several components has to be managed, a switch is introduced. The reconfiguration strategy which is chosen for each switch is modeled by a Moore machine, associated to this switch. Two kinds of switches are used in this study case. Switches of type *M3* express a strategy "1 out of 2, latest replacement, latest resumption" (replacement occurs when the main component fails if the spare component is available, and resumption occurs when the spare fails if the main is available); the switch of type *M2* expresses a strategy "2 out of 3, latest replacement, latest resumption". The corresponding Moore machines are given in Figure 5c.



a) Structure of the GBDMP



b) SMP associated to a leaf of type *SF* (top) and *F* (bottom)



c) Moore machines associated to switches of type *M2* (top) and *M3* (bottom)

Figure 5. GBDMP of the controlled coolant feeding system.

Finally, a control equipment in charge of executing the reconfiguration is associated to each switch (leaves R_A to R_E depicted by grey boxes at Figure 3a). A failure of the switch controller implies the loss of reconfiguration ability.

5.3 *Comparative study*

The application of Algorithm 1 to the GBDMP of Figure 5 gives the result $\mathcal{L}_1 = \mathcal{L}_{MCS}(\Subset)$. A selection of elements of this language is reported on Table 1[3].

Total number of MCS	Selection of MCS
Length 2: 14	$f_a^{Ex1}\ f_a^{Ex2}$ $f_a^{TEa1}\ f_a^{TEa2}$ $\boldsymbol{f_a^{RB}\ f_a^{Pr1}}$ $\boldsymbol{f_a^{RD}\ f_a^{Grid}}$
Length 3: 37	$F_p^{Di1}\ f_a^{Grid}\ f_a^{Di2}$ $f_a^{TEa1}\ f_a^{TEb2}\ f_a^{Di2}$ $\boldsymbol{f_a^{RE}\ f_a^{Grid}\ f_a^{Di1}}$ $\boldsymbol{f_a^{RC}\ f_a^{RD}\ f_a^{TEb1}}$
Length 4: 93	$f_a^{Tr1}\ f_p^{Di1}\ f_a^{Tr2}\ f_a^{TEa2}$ $\boldsymbol{f_a^{Ex1}\ r_p^{Ex1}\ f_a^{Ex3}\ f_a^{Ex2}}$ $\boldsymbol{f_a^{Pr1}\ f_a^{RB}\ r_p^{Pr1}\ f_a^{Pr2}}$ $\boldsymbol{f_a^{Pr1}\ r_p^{Pr1}\ f_a^{RB}\ f_a^{Pr2}}$
Length 5: 240	$f_a^{Tr1}\ f_a^{TEb1}\ f_a^{Di1}\ f_p^{Di2}\ f_a^{Tr2}$ $\boldsymbol{f_a^{Grid}\ r_p^{Grid}\ f_a^{RD}\ f_p^{Di2}\ f_a^{Di1}}$ $\boldsymbol{f_a^{Tr1}\ f_a^{RE}\ f_a^{RD}\ r_p^{Tr1}\ f_a^{Tr2}}$ $\boldsymbol{f_a^{Grid}\ f_a^{Di1}\ r_p^{Grid}\ f_a^{TEa2}\ f_a^{TEb1}}$
Length 6: 1560	$\boldsymbol{f_a^{RE}\ f_a^{TEb1}\ f_a^{Di1}\ r_p^{TEb1}\ f_p^{Di2}\ f_a^{Tr1}}$ $\boldsymbol{f_a^{Grid}\ f_a^{Di1}\ r_p^{Grid}\ f_a^{TEa2}\ f_a^{RE}\ f_a^{Tr1}}$ $\boldsymbol{f_a^{Tr1}\ f_a^{Tr2}\ f_a^{RD}\ f_a^{Di1}\ r_p^{Tr1}\ f_a^{Di2}}$ $\boldsymbol{f_a^{TEa1}\ f_a^{Tr1}\ r_p^{TEa1}\ f_a^{Tr2}\ f_p^{Di1}\ f_a^{TEa2}}$

Table 1. Extract of $\mathcal{L}_{MCS}(\Subset)$ for the coolant feeding water

This result has been produced in 45 minutes on a standard laptop (2,9 GHz). The first 200 sequences were found in less than one minute. Moreover, all cut sequences of length inferior or equal to 5 (resp. 6) were found in less than 5 minutes (resp. 55 minutes).

This case study has been addressed differently in [Chaux2012]. Firstly the system was modeled using BDMP and not GBDMP (the modifications correspond to the grey elements on Figure 5). Secondly, the definition of MCS was based on $\subseteq$ instead of $\Subset$.

Let us compare $\mathcal{L}_1$ with $\mathcal{L}_2 = \mathcal{L}_{MCS}(\subseteq)$, computed from the BDMP model. Bold sequences in Table 1 do not belong to $\mathcal{L}_2$; moreover some sequences of length 5 and 6 belong to $\mathcal{L}_2$ but not to $\mathcal{L}_1$ (of course they are not displayed in this table).

One of the most important improvement from BDMP to GBDMP is the replacement of the *trigger* primitive by the *switch* one. Both translate a reconfiguration mechanism in the model, but a trigger refers to a unique strategy of switching that may not fail [Piriou2016]. On the other hand, in GBDMP, the Moore machines associated to the switches allow to describe multiple reconfiguration strategies that may fail. Hence the sequences that are not in the intersection of $\mathcal{L}_1$ and $\mathcal{L}_2$ are explained by at least one of the three following reasons:

1. $\Subset$ is less restrictive than $\subseteq$ for the MCS calculus.
2. In the GBDMP model, the failure of reconfigurations are taken into account (through grey leaves R_A to R_E), what implies a new dysfunctional behavior.
3. The reconfiguration strategies included in the grey switches (*S1* to *S6*) differ from the unique strategy translated by a BDMP trigger, what implies a different dysfunctional behavior.

To illustrate the differences explained by the first reason, let us consider the sequence:

[3] *f* and *r* mean *failure* and *repairs*; *a* and *p* mean *active* and *passive*. Then f_p^{Tr1} represents the event: *failure of the leaf Tr1 while it is passive (mode 0)*.

$f_a^{Grid}\ f_a^{Di1}\ r_p^{Grid}\ f_a^{TEa2}\ f_a^{TEb1}$.
It is not a MCS according to the relation $\subseteq$ because it includes the MCS $f_a^{Grid}\ f_a^{Di1}\ f_a^{TEa2}$. However, it is a MCS according to the relation $\Subset$ because:
$[f_a^{Grid}\ f_a^{Di1}\ f_a^{TEa2}]$ = {Grid, Di1, TEa2}, whereas
$[f_a^{Grid}\ f_a^{Di1}\ r_p^{Grid}\ f_a^{TEa2}\ f_a^{TEb1}]$ = {Di1, TEa2, TEb1}.

The differences coming from the modeling of reconfigurations failures can be illustrated by the sequences $f_a^{RB}\ f_a^{Pr1}$ and $f_a^{Pr1}\ f_a^{RB}\ r_p^{Pr1}\ f_a^{Pr2}$ that are respectively symptomatic of the failure of the *replacement* and the *resumption* of the pump Pr1.

Let us now exemplify the differences explained by the third reason using two sequences:

- $f_a^{Ex1}\ r_p^{Ex1}\ f_a^{Ex3}\ f_a^{Ex2} \in \mathcal{L}_1 \setminus \mathcal{L}_2$. Indeed, according to the GBDMP model, the resumption of Ex1 does not occur after its own repairs. Then $f_a^{Ex3}\ f_a^{Ex2}$ can occur after that. On the contrary, with the BDMP model, Ex1 is resumed just after it is repaired and Ex3 is put in standby mode, the event f_a^{Ex3} is consequently not possible. Then the considered sequence is not even in the evolution language $\mathcal{L}_{\mathcal{E}}$ of the model.
- $f_a^{TEa1}\ f_a^{TEb1}\ r_p^{TEa1}\ \boldsymbol{f_p^{Di2}}\ f_a^{Tea1} \in \mathcal{L}_2 \setminus \mathcal{L}_1$. Indeed, according to the GBDMP model, the resumption of power supply on the first line following to the repairs of TEa1 is not immediate (contrarily to what describes the BDMP model). Then Di2 may fail after that only while it is active, what would result in the sequence $f_a^{TEa1}\ f_a^{TEb1}\ r_p^{TEa1}\ \boldsymbol{f_a^{Di2}}\ f_a^{Tea1}$ which is not minimal because:
  $f_a^{TEa1} f_a^{TEb1} f_a^{Di2} \Subset f_a^{TEa1} f_a^{TEb1} r_p^{TEa1} f_a^{Di2} f_a^{Tea1}$.

Finally, some differences between $\mathcal{L}_1$ and $\mathcal{L}_2$ are explained by a combination of the three above-mentioned reasons. For example, the sequence $f_a^{Pr1}\ r_p^{Pr1}\ f_a^{RB}\ f_a^{Pr2} \in \mathcal{L}_1 \setminus \mathcal{L}_2$ because of the three reasons.

6 CONCLUSIONS

In this paper a new definition of MCS has been proposed to perform a more accurate qualitative safety analysis for dynamic repairable and reconfigurable systems. This definition is suitable for any dynamic system. Moreover an algorithm to compute the MCS set of a system modeled in GBDMP has been proposed in order to take advantages of the new definition. The comparative study performed shows the benefits of this approach: the model is more precise and the analysis results are more relevant.

Nevertheless, even if the qualitative analysis supplies relevant information on the system dysfunctional behavior, it is not enough to validate that a design meets its safety requirement. Indeed a short MCS can be highly improbable. To combine the MCS calculus with the probability assessment of the system failure is a promising idea to improve the relevance of safety analysis. [Brameret2015] defined a factor to order the states of a Markov chain according to their probabilistic relevance. Given that a GBDMP model describes a Markov chain by intention, the Algorithm 1 can be improved by introducing a heuristic based on the probabilistic relevance factor to drive the state space exploration.

REFERENCES

[Bouissou2003]: Bouissou M., Bon J.L. *A new formalism that combines advantages of fault trees and Markov models: Boolean logic Driven Markov Processes.* In Reliability Engineering & System Safety. 2003, vol. 82 (2), pp. 149-163.

[Brameret2015]: Brameret, P.-A., Rauzy, A., and Roussel, J.-M. *Automated generation of partial Markov chain from high level descriptions*. In Reliability Engineering & System Safety. 2015, vol. 139, pp. 179-187.

[Chaux2012]: Chaux, P.-Y., Roussel, J.-M., Lesage, J.-J., Deleuze, G., and Bouissou, M. *Systematic extraction of minimal cut sequences from a BDMP model*. In Proc. 21th European Safety & Reliability Conference (ESREL'12). 2012, Helsinki (Finland), 8 pages.

[Chaux2013]: Chaux, P.-Y., Roussel, J.-M., Lesage, J.-J., Deleuze, G. & Bouissou M. *Towards a unified definition of minimal cut sequences*. In IFAC Dependable Control on Discrete event Systems. 2013, York (UK). 6 pages.

[Meduna2012]: Meduna, A. *Automata and languages: theory and applications*. 2012, Springer

[Piriou2014]: Piriou, P.-Y., Faure, J.-M. & Lesage J.-J. *Control-in-the-loop Model Based Safety Analysis*. In European Safety & Reliability Conference (ESREL'14). 2014, Wroclaw (Poland).

[Piriou2016]: Piriou, P.-Y., Faure, J.-M. & Lesage J.-J. *Generalized Boolean logic Driven Markov Processes: a powerful modeling framework for MBSA of dynamic repairable and reconfigurable systems*, paper submitted to Reliability Engineering & System Safety, 2016.

[Rauzy2011]: Rauzy, A. *Sequence Algebra, Sequence Decision Diagrams and Dynamic Fault Trees*. Reliability Engineering & System Safety. 2011, vol. 96 (7), pp. 785-792.

[Tang2004]: Tang, Z., Dugan, J.B. *Minimal cut set/sequence generation for dynamic fault trees*. In Annual Reliability and Maintainability Symposium (RAMS). 2004, Los Angeles (USA).

[Walker2007]: Walker, M., Bottaci, L., and Papadopoulos, Y. (2007). *Compositional temporal fault tree analysis*. In Proc. 26th International Conference on Computer Safety, Reliability, and Security. 2007, Nurmberg (Germany).