Finding the Minimal Cut Sequences of dynamic, repairable and reconfigurable systems from GBDMP models

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

Finding the Minimal Cut Sequences of dynamic, repairable and reconfigurable systems from GBDMP models

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

Minimal Cut Sequences (MCS) 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 these systems are both repairable and reconfigurable. A new definition for this class of systems as well as an algorithm to compute these sequences from a safety analysis model, in the form of a GBDMP (Generalized Boolean logic Driven Markov Processes) model, are then proposed. These contributions are illustrated on a case study from power industry. Comparison of the obtained MCS to those which are yielded by algorithms based on the previous definitions permits to highlight the relevance of the approach.

I - INTRODUCTION

Qualitative safety analysis is aiming at finding the causes of the failure of the system under study, whatever the failure rates of its components which are considered only for quantitative analysis ([Ge 2014], [Li 2015]). For dynamic systems, these causes are described by *cut sequences* (CS) [Tang 2004], sequences of event occurrences that lead the system from its initial state to a failure state without passing through another failure state. Since the set of CS is very huge for real critical systems, the set of *Minimal Cut Sequences* (MCS), subset of CS that is sufficient to represent all CS, must be searched. Several formal definitions of MCS can be found in the literature. Some of them ([Tang 2004], [Walker 2007], [Rauzy 2011]) assume that the components of the system are non-repairable. A formal definition of MCS for systems whose components are repairable can be found in [Chaux 2013]. The case of repairable and reconfigurable systems is addressed in none of these worthwhile contributions, however.

Modern systems are indeed expected to be more and more flexible and dependable. To meet these objectives, 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 a mission), fault tolerance objectives (management of redundant resources), maintenance policies or production needs. These reconfiguration strategies impact strongly system

safety. In particular, it has been shown in [Piriou 2014], on the basis of a representative case study, that 94 **new** MCS whose length is smaller or equal to five (10 sequences of length 2, 54 sequences of length 4 and 30 sequences of length 5) are found when the failures of the control system that manages the reconfiguration strategies are considered, while only 84 MCS were detected when only the failures of the process elements were considered.

The first objective of this paper is to show, by using counter-examples, that the definitions of MCS which have been previously published are not suitable for dynamic, repairable and reconfigurable systems. A new formal definition is afterwards proposed. This definition relies on the postulate that a dysfunctional sequence (CS or MCS) is characterized both by the order of the events occurrences it includes and by the set of faulty components at the end of this sequence.

Once this definition stated, an algorithm for computing MCS from fault-forecasting models built by using Generalized Boolean logic Driven Markov Processes (GBDMP) [Piriou 2017], an appropriate modeling framework for safety analysis of repairable and reconfigurable systems, is developed. A case study illustrates both the GBDMP modeling power and the MCS computing principle. The set of MCS obtained is afterwards compared to the ones obtained by using the previous definitions of MCS and the differences are highlighted and discussed.

The outline of the paper is the following one. In the next section, two existing definitions of MCS are recalled. Section 3 proposes a new definition which allows to deal with dynamic repairable and reconfigurable systems. Section 4 shows how the MCS set can be computed from a GBDMP model. A case study is addressed in section 5 to illustrate the approach. Finally, concluding remarks and perspectives are drawn up in section 6.

II - BACKGROUND

Two formal definitions of MCS that have been recently proposed are reminded in this part. Both definitions rely on languages theory. This explains why the problem will be formally stated in the next sub-section by using this modeling framework [Meduna 2012]. Symbols definition are given on the go and are reported in Appendix 1.

2.1 Problem statement

In an informal manner, the set of MCS is the minimal set of sequences of minimal length that is necessary and sufficient to represent a whole set of cut sequences. To propose a formal definition of this set, the following concepts and notations are introduced:

- Σ is the alphabet of events that must be considered for safety analysis; this set includes obviously failure and repair events, which will be noted respectively f_i and r_i for the i^{th} component, but may include also events that represent a normal behavior, like phase change events.
- A sequence of event occurrences is noted σ. Σ^* is the set of all sequences of event occurrences which can be built on Σ. The zero-length sequence is noted ε.
- A language $\mathcal{L} \subseteq \Sigma^*$ built on Σ is a set of event occurrences sequences.
- A finite automaton $\mathcal{A}$ is a 5-tuple:
  $< \Sigma, Q, q_0, Q_M, \delta >$, where:
  - Σ is a finite set of events;
  - Q is a finite set of states;

- $q_0 \in Q$ is the initial state[1];
- $Q_M \subseteq Q$ is the set of the marked states, states that usually represent the end of an evolution;
- δ is the transition function: $Q \times \Sigma \rightarrow Q$

- Two regular languages can be defined from a finite automaton $\mathcal{A}$:
  - $\mathcal{L}(\mathcal{A}) = \mathcal{L}_E \subseteq \Sigma^*$, the evolution language that contains all sequences of events occurrences that correspond to paths from the initial state to any state;
  - $\mathcal{L}_m(\mathcal{A}) = \mathcal{L}_M \subseteq \mathcal{L}_E$, the marked language that contains all sequences of events occurrences that correspond to paths from the initial state to a marked state.
- For safety analysis, the marked states will represent global failure states. Therefore, the marked language $\mathcal{L}_M$ can be termed failure language and noted $\mathcal{L}_F$.
- $\mathcal{L}_{CS} \subseteq \mathcal{L}_F$ is the set of cut sequences:

  $\mathcal{L}_{CS} = \{ \sigma \in \mathcal{L}_F \mid \forall \sigma' \in \text{Pref}(\sigma) \wedge \sigma' \neq \sigma, \sigma' \notin \mathcal{L}_F \}$,

  where Pref(σ) is the set of all prefixes of σ; if f_1 and f_2 are two failure events, r_1 a repair event and $\sigma = f_1 f_2 r_1$ for instance, Pref(σ)={ε, f_1,f_1f_2, $f_1f_2r_1$}
- $\mathcal{L}_{MCS}(\mathcal{R}) \subseteq \mathcal{L}_{CS}$ is the set of Minimal Cut Sequences according to an order relation $\mathcal{R}$:

  $\mathcal{L}_{MCS}(\mathcal{R}) = \{ \sigma \in \mathcal{L}_{CS} \mid \forall \sigma' \in \mathcal{L}_{CS},\ \sigma\ \mathcal{R}\ \sigma' \Rightarrow \sigma = \sigma' \}$

The last definition means that a cut sequence σ is minimal if it represents every cut sequence σ' which is related to σ by an order relation $\mathcal{R}$. This order relation $\mathcal{R}$ depends on the definition of a MCS, as it will be shown in the sequel of this section.

To illustrate these definitions, let us consider a simple example: two components A and B in standby redundancy which is driven by a control component C. The finite automaton $\mathcal{A}$ of Figure 1 models the dysfunctional behavior of this system[2], assuming that A, B and C are repairable and, in the initial state q_0, A is active and B inactive. For this automaton: $\Sigma = \{f_A, r_A, f_B, r_B, f_C, r_C\}$, $\mathcal{L}_E = \mathcal{L}(\mathcal{A})$, $\mathcal{L}_F = \mathcal{L}_m(\mathcal{A})$ and $\mathcal{L}_{CS} = \mathcal{L}_m(\mathcal{A}')$, where $\mathcal{A}'$ is obtained from $\mathcal{A}$ by removing every transition that leaves a marked state.

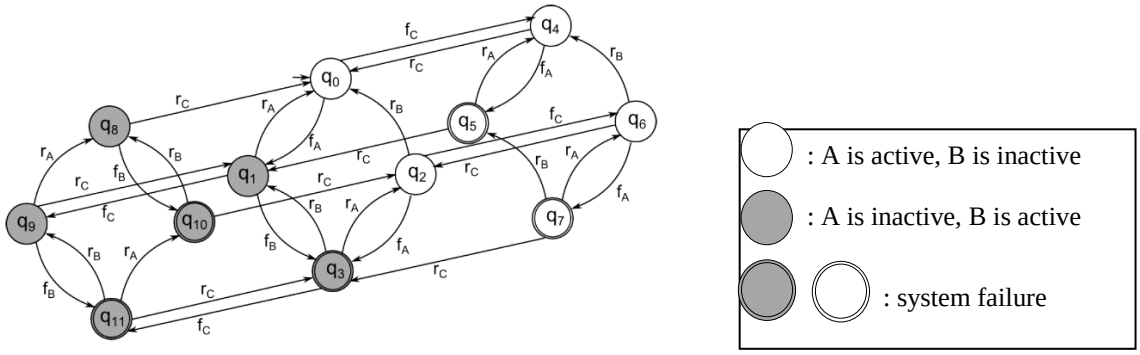


Figure 1. Finite automaton modeling two components A and B in standby redundancy driven by a control component C.

[1] It will be assumed in the rest of this paper that the initial state is unique.

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

Last, for any evolution sequence $\sigma \in \mathcal{L}_E$, it is possible to define its *covering cut* [σ], 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}, for instance.

2.2 First basic definition of MCS

The first proposition for $\mathcal{R}$ is the *sequence inclusion* ([Tang 2004], [Chaux 2012]), which will be denoted "$\subseteq$": a sequence σ is included into another sequence σ' if and only if all events of σ are present in the same order in σ'. This definition has been proposed for non-repairable systems and therefore is appropriate only for this class of systems. For the automaton $\mathcal{A}''$ obtained from $\mathcal{A}$ by removing all transitions[3] labelled with a repair event r_i, the set of MCS is: $\mathcal{L}_{MCS}(\subseteq)$={$f_A f_B$, $f_C f_A$}; the system fails when A and B are faulty (states q_3 and q_{11}) or when C and A are faulty while C have failed before A (state q_5).

This definition of MCS is no more adequate when repairable systems are considered. If we focus for instance on the sequence $f_A f_C r_A f_B$ which is possible in $\mathcal{A}$ and corresponds to the shortest path from q_0 to q_{10}, this sequence is not a minimal cut sequence according to the definition of the sequence inclusion relation, because it includes the minimal cut sequence $f_A f_B$ ($f_A f_B \subseteq f_A f_C r_A f_B$). However, this sequence $f_A f_C r_A f_B$ is not represented by $f_A f_B$ because only the first one describes the particular dysfunctional behavior: when B fails after the sequence *failure of A* then *failure of C* then *repair of A*, A cannot be activated, while faultless, because C is currently faulty.

2.3 A definition of MCS for repairable systems

A promising definition of MCS for dynamic and repairable systems is proposed in [Chaux 2013]. It is based on the concept of *coherent dynamic system*. A system is termed *coherent* if and only if it is possible to build a new failure sequence from any failure sequence ($\sigma \in \mathcal{L}_F$) by adding events to σ according to the following two operations:

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

If we note 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 to σ by using the operations 1 and 2, f_1 and f_2 can be extended to languages:

$\forall \mathcal{L} \subseteq \mathcal{L}_E$, $f_1(\mathcal{L})=\bigcup_{\sigma \in \mathcal{L}} f_1(\sigma)$, $f_2(\mathcal{L})=\bigcup_{\sigma \in \mathcal{L}} f_2(\sigma)$.

With these notations, a *coherent system* can be then formally defined as a system that verifies the following property:

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

In such a system, a sequence σ represents another sequence σ' if and only if σ' can be built from σ by using the operations 1 and 2. Hence, a new definition of the relation $\mathcal{R}$ arises: the *coherence relation*, which will be noted ⊨:

[3] Some states, like the marked states q_7 and q_{10}, become not reachable when these transitions are removed.

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

For the example depicted at Figure 1, with definition (2), the set of MCS is: $\mathcal{L}_{MCS}(\vDash)=\{f_A f_B, f_C f_A, f_A f_C r_A f_B\}$ what is consistent.

III - DEFINING THE MCS FOR REPAIRABLE AND RECONFIGURABLE SYSTEMS

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 that includes two repairable components A and B and performs a mission in two phases. During the first phase, A and B are in standby redundancy (in the initial state, A is active and B inactive), whereas during the second phase both are required. In this figure, φ represents the event of phase switching, whatever the source and destination phases.

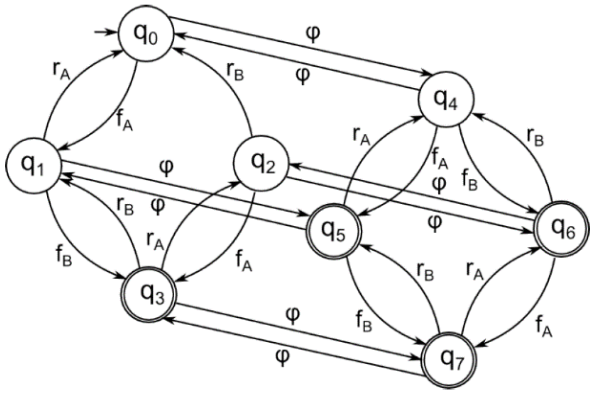


Figure 2. Automaton modeling two components A and B that perform a phased mission.

This system is not coherent according to (1). If φf_A is the considered failure sequence indeed (the component A fails during the second phase), the sequence $\varphi f_A \varphi$ which is obtained by adding at the end of the first sequence an occurrence of φ is no more a failure sequence (only B is sufficient during the first phase), although the covering cut remains unchanged ($[\varphi f_A] = [\varphi f_A \varphi] = \{A\}$).

The definition of MCS proposed at 2.3 is therefore not appropriate for repairable and reconfigurable systems.

3.2 A new minimality criterion for CS

To solve this issue, we argue that, for dynamic, repairable **and reconfigurable** systems, the set of MCS must be based on a relation which is the conjunction of two other relations:

- *sequence inclusion*, because the system failure depends on the relative order of components failure event occurrences in dynamic systems;
- *covering cut inclusion*, because the state of the system depends also on the states (faultless/faulty) of its components, too.

This new relation can be named *covering cut and sequence inclusion* and will be noted $\Subset$. It is formally defined as follows:

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

where σ and σ' are sequences of event occurrences and $\mathcal{L}_E$ the evolution language of the automaton.

It must be underlined that, as the two operations 1 and 2 defined at 2.3 neither modify the order of the event occurrences of the initial sequence nor remove any component of its covering cut, (3) is less restrictive than the *coherence relation* (2). If (2) holds, (3) mandatorily holds, as formalized in (4) with the previous notations, but the opposite is not true.

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

The relation $\Subset$ is generic and must be selected to determine the set of MCS whatever the considered system (only repairable or repairable **and** reconfigurable). This highlights the relevance of our approach and will be illustrated on the previous two examples:

- for the example of Figure 1, which is a repairable but not reconfigurable system:
  $\mathcal{L}_{MCS}(\Subset) = \mathcal{L}_{MCS}(\vDash) = \{f_A f_B, f_C f_A, f_A f_C r_A f_B\}$,
- for the example of Figure 2, which is a repairable and reconfigurable system:
  $\mathcal{L}_{MCS}(\Subset) = \{f_A f_B, f_A \varphi, \varphi f_A, \varphi f_B\}$.

It must be noted that the relation (3) provides a correct set of MCS even for the second example, which was not the case for the relation (2). Automatic computation of the set of MCS is addressed in the next section.

IV - COMPUTING THE SET OF MCS FROM A GBDMP MODEL

4.1 Recall on GBDMP

Generalized Boolean logic Driven Markov Processes (GBDMP) is an extension for reconfigurable systems of the BDMP framework defined in [Bouissou 2003]. This framework has been originally developed for safety analysis of systems whose lifespan is very long, like power production and distribution systems, and consequently must be repairable during operation. Since this date, it has been also used for security analysis of these systems [Pietre-Cambacedes 2011], [Kriaa 2012]. GBDMP is a proposal to take benefit of the strengths of the BDMP framework and to remove some of its limitations, in particular for analysis of reconfigurable systems.

Basically, a GBDMP model integrates three models:

- a representation of the structure of the system, in the form of an extended fault tree;

- the description by Switched Markov Processes (SMP) of the behaviors of the components of the system;
- the modeling of reconfiguration mechanisms, with Moore machines.

The bases of GBDMP syntax and semantics are now briefly given and exemplified. They are formally and broadly presented in [Piriou 2017].

Definition 1. A Generalized Boolean logic Driven Markov Process is a 6-tuple $<V, E, K, \upsilon, str, smp>$ where:

- $V = N \cup S = G \cup L \cup S$ *is a set of vertices that is composed of two disjointed sets: the set of nodes N and the set of switches S; the set of nodes N is itself composed of the set of gates G and the set of leaves L, with* $G \cap L = \varnothing$.
- $E = E_F \cup E_S$, *with* $E_F \cap E_S = \varnothing$, *is a set of directed edges, such that:* $E_F \subseteq G \times N$ *and* $E_S \subseteq (N \times S) \cup (S \times N)$.
- $K: G \rightarrow \mathbb{N}^*$ *is a function that define the kind of gate. This function is the same as the one used in BDMP (see [Bouissou 2003]).*
- $\upsilon: E \rightarrow \mathbb{N}$ *is a function that associates an integer label to every edge.*
- $str: S \rightarrow \mathbb{M}$ *is a function that associates a Moore machine (which represents a reconfiguration strategy) to every switch.* $\mathbb{M}$ *designates the set of Moore machines.*
- $smp: C \rightarrow \mathbb{P}$ *is a function that associates a SMP to every 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 an extended fault tree (part a) of Figure 3) which 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 by 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 labels of these edges are given by the function υ and represent respectively the operation mode of the destination leaf and the number of the input or output of the Moore machine associated to the destination/origin switch.

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 3b. The reconfiguration strategy which is implemented in the switch S1 is modelled by the Moore machine of Figure 3c.

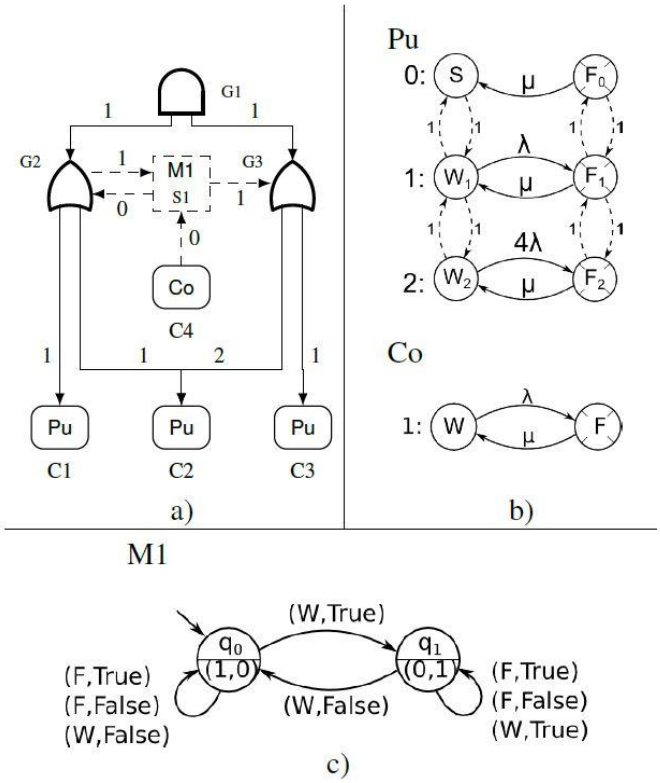


Figure 3: Example of GBDMP. a) Structure modeling; b) SMP Pu (associated to C1, C2 and C3) and Co (associated to C4); c) Moore machine M1 (associated to S1)

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 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 to represent a component with two working modes (chains 1 and 2) and one standby mode (chain 0); 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 four times greater than the corresponding rate in the first working mode. The transitions between two states of two different chains of a k-SMP (dashed arrows) correspond to operation mode changes. If no failure on-demand occurs when the operation mode is changed, the label of the transition is equal to 1 (case of Figure 3 b); if this is not the case, the transition is labeled by the corresponding failure/success rate that belongs to [0, 1].

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. For example, let q_0 be the active state in the Moore machine M1 of Figure 3 c. In this state, the outputs of this machine are respectively *True* (1) and *False* (0), what means that the gate G2, connected to the output #0, is required and the gate G3, connected to the output #1, not required. Hence, C1 and C2 are activated in the operation mode 1 and C3 is deactivated, because it is not then required. The transition between q_0 and q_1 is fired when the associated condition "(W, True)" is true, i.e. if the state of the SMP of C4 (input #0 of S1) is W and if the status of the gate G2 (input #1 of S1) is True, what means that this gate is faulty. Firing this transition

changes the active state which becomes q_1 and consequently the outputs values: G2 is no more required but G3 is required now. Hence, C1 is deactivated, C2 is activated in the operation mode 2 and C3 is activated in the operation mode 1, according to the labels of the edges of E_F.

4.2 Translation of a GBDMP into a finite state automaton.

As qualitative analysis does not consider probabilistic values, the transitions of every SMP must be labeled by failure/repair or mode change events, and not by failure/repair or failure on demand rates before this translation. Once this simple modification made, a well-formed[4] GBDMP $<V, E, K, \upsilon, str, smp>$ can always be translated into a finite state automaton $<\Sigma, Q, q_0, Q_M, \delta>$, with the notations introduced at section 2.1. It must be noted that this automaton is non-timed, i.e. that only logical time (order of the event occurrences) is considered. This is not an issue for qualitative analysis because this analysis does not consider physical time (time between the occurrences). If the designer wants to introduce a timed event, e.g. mode change of a pump every week/month, he/she must merely define a non-timed event which corresponds to this request to change the mode.

The evolutions of a GBDMP model are driven by two types of events:

- *spontaneous* events. Such an event is the origin of an evolution. Failure events (except failure on-demand), repair events, phase change events are examples of spontaneous events. They correspond to the solid arrows in the SMP representation (Figure 3 b)).
- *provoked* events. A provoked event is the direct or indirect consequence of at least one spontaneous event. Operation mode changes, e.g. from standby to working, and failure on-demand are examples of provoked events. These events correspond to the dashed arrows in the SMP representation.

Hence, the alphabet Σ of the automaton is the disjunction of the sets of these two kinds of events.

The global state of a GBDMP model is fully defined by the local states of every SMP and Moore machine. In particular, from the knowledge of the initial states of the Moore machines, it is possible to determine the initial state of every SMP, what gives the initial global state q_0. Then, the set of states Q is the set of combinations of these local states which are reachable from q_0 by using sequentially the transition function δ with the two types of events. Finally, the marked states are the states where the status of the gate which corresponds to the top event, or undesirable event, of the extended fault tree is *True*.

4.3 An algorithm to obtain the set of MCS from a GBDMP model.

Algorithm 1, given at Appendix 2, computes on the fly the set of MCS by performing a breadth-first exploration of the state space Q. Thus the sequences are computed from the shortest one (empty sequence ε) to the longest ones. At each iteration of the "while loop" (lines 6-23), the set of sequences of length k (*seqOfCurLength*) is determined from the set of sequences of length $k-1$ (*seqOfLastLength*) obtained at the previous iteration (line 3

[4] Five syntactic properties that define a well-formed GBDMP are given in [Piriou 2017].

initializes this set at { } for performing the first iteration). The breadth-first exploration ensures that, for each new cut sequence found, it is possible to determine whether it is minimal or not by comparison with the already found MCS (lines 14-16). This algorithm uses the function $\mathcal{E}$: $Q \longrightarrow (\Sigma)$ which gives the set of events that may occur in a given state of the GBDMP to limit the analysis to the only events that can change the active state.

The exploration is stopped if one of the following two conditions is met (lines 12-13):

1. A marked state is reached, because a cut sequence leads compulsorily to a failure state.
2. A state which has been already visited is met another time, what means that the current sequence includes a loop and can be represented by a shorter sequence by removing this loop.

The theoretical complexity of Algorithm 1 is $\mathcal{O}(Card(Q)!^2)$, which is very high. But the 2-occurrences long sequences are first computed, then the 3-occurrences long sequences and so on[5]. Computation time obviously increases with the length but, as the shortest sequences are generally the most critical ones, whatever the failure rates, this approach provides in a reasonable time useful qualitative results, even if there are not complete. Then, this algorithm can be applied to real systems because, in practice, the aim of qualitative analysis is not to compute the whole set of MCS but to determine the shortest MCS.

Last, it must be noted that this algorithm can be easily selected for any other definition of MCS, by replacing the relation ⋐ at line 15 by the corresponding relation.

V - EXAMPLE

5.1 System description

This example was proposed by the French company Electricité de France, which designs and operates power plants and has partially funded this research. A power plant includes indeed several highly critical repairable and reconfigurable systems that require an accurate safety analysis before operation. Moreover, detailed information on the structure of the plant, its reconfiguration strategies and the behavior of its components were available, which is mandatory for careful modeling. The aim of this section is therefore to show that our approach can be applied to real industrial systems and to highlight its benefits by discussion of the results it provides on the basis of a representative example: the Pools Cooling system, one of the most critical systems of a nuclear power plant.

A nuclear power plant is a phased mission system whose mission globally comprises four phases:

- *Power Production* (PP),
- *Stopped for Refueling* (SR) where the used fuel is unloaded and new fuel loaded,
- *Transient Phases* (TP_1 and TP_2), phases between SR (PP) and PP (SR) during which the power is increased (decreased).

[5] *Card(Q)* is the cardinality of the set of states Q and also the maximum length of the cut sequences.

Within such a plant, the Pools Cooling system is a particularly critical system which performs three functions: Fuel Pool Cooling (FPC), Reactor Pool Cooling (RPC) and Passive redundancy of the Primary Circuit Cooling (PPCC) which is another cooling system not detailed in this paper, for room reasons. The first function is required for every phase whereas the second one is required only during the SR phase, when the reactor pool is filled in because the reactor is open, and the third one only during the transient phases. This system includes (Figure 4) two pumps P1 and P2, two manual valves V1 and V2 and two heat exchangers HE1 and HE2. The pumps P1 and P2 are designed to ensure respectively, when faultless, the FPC function and the RPC or PPCC function.

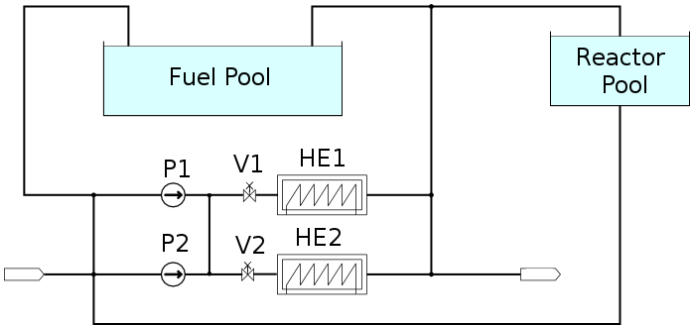


Figure 4: Layout of the Pools Cooling system

Five reconfiguration strategies (RS1 to RS5) are defined textually for this system; the first three ones are global and the last two ones focus on P1 and P2:

- The RPC function is required only during the SR phase (RS1) and the PPCC function only during the transient phases (RS2).
- When the Primary Circuit Cooling system fails during a transient phase, it is replaced by the Pools Cooling system (RS3).
- When P1 (P2) fails, it must be deactivated. Else, it must be activated as long as the FPC (RPC or PPCC) function(s) is (are) required. When P2 (P1) fails while the RPC or PPCC (FPC) function(s) are (is) required, P1 (P2) is switched in the *Overspeed* operation mode to ensure the load of the failed pump (RS4 and RS5).

These strategies will be formally described by Moore machines associated to switches of the GBDMP model in what follows. Their implementation on the control architecture is given at Figure 5 which includes five PLC (Programmable Logic Controllers) and two redundant buses. Two redundant instances of the codes that are based on each strategy (RSi and RSi') are implemented on two different controllers, for safety reasons.

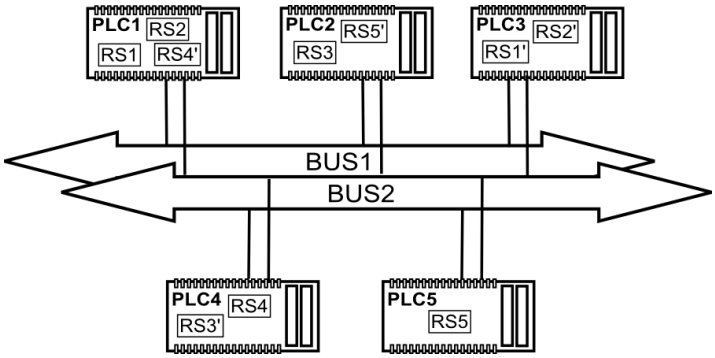


Figure 5: Operational control architecture

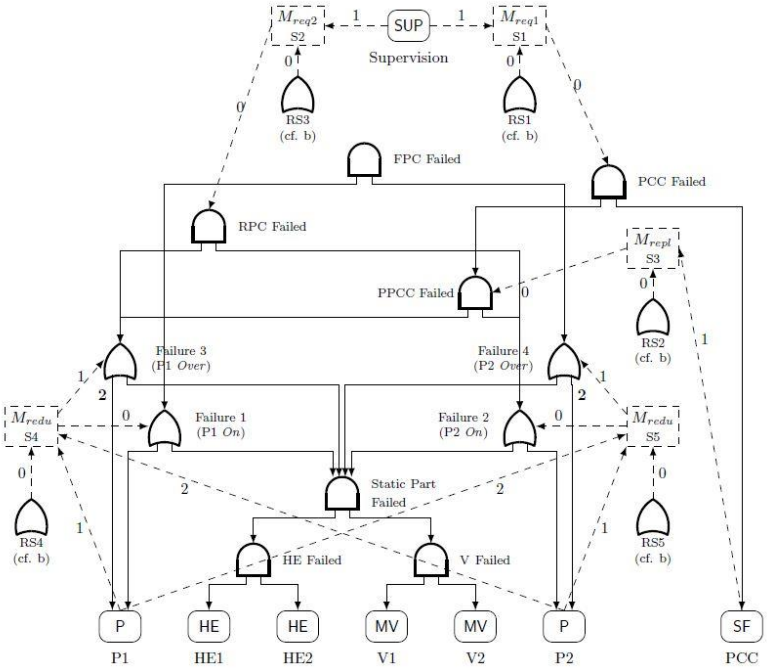
5.2 GBDMP model

The extended fault tree that is derived from the functional and hardware structure of the system is depicted at Figure 6 (part a) for the process and b) for the control architecture). The SMP which model the behavior of the leaves of this tree are given at Figures 7-12 whilst the Moore machines that formally describe the reconfiguration strategies are shown at Figure 13.

5.2.1 Extended fault tree

It must be noted that this model is represented in two parts for clarity reasons but that these parts are connected by the gates RS_i, which are common to both parts and represent the failures of the implementation of the reconfiguration strategies. The failures of the control components impact indeed the global safety.

a)



b)

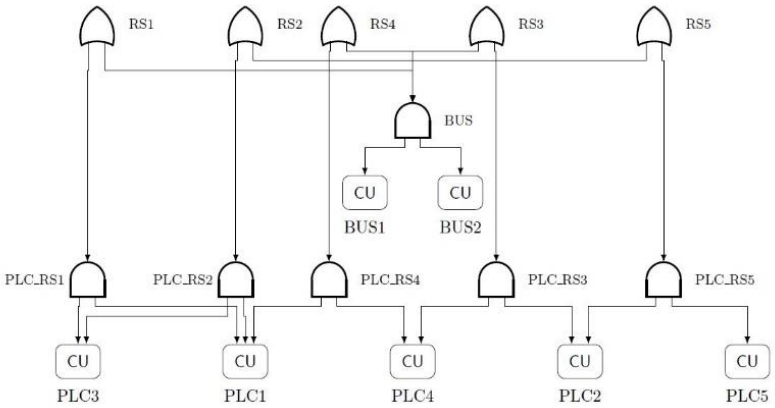


Figure 6: Structural view of the system's GBDMP model

Figure 6a focuses on the failures of the three functions FPC, RPC and PPCC. By considering only the static gates, FPC, for instance, fails when both Failure1 and Failure4 have occurred. Failure1 occurs when the pump P1 has failed in mode On or when both heat exchangers and both valves have failed too. Failure4 occurs when the pump P2 has failed in mode Over or when both heat exchangers and both valves have failed too.

This figure comprises, besides the static gates, 5 switches (S1 to S5) that clearly point out that a reconfiguration is performed when a part of the system fails. The dynamic behaviors of switches are represented by Moore machines which are given at section 5.2.3. The inputs/outputs of the switches (represented in the figure by incoming/outgoing dashed arrows) are variables which model the states of leaves or gates of the tree. The relations between these state variables and the evolutions of the Moore machines that describe the behavior of the switches is exemplified at Appendix 3.

Figure 6b describes the failures of the implementation of the reconfiguration strategies and is easier to interpret because it comprises only static gates. The implementation of a reconfiguration strategy (RS1 for instance) fails either when both buses of the control architecture fail or both PLC where an instance of the code based on this strategy (PLC1 and PLC3 in this case) is implemented fail.

5.2.2 Models of the leaves

The behavior of each pump is modeled by a 3-SMP P with three operation modes: Stopped (0), On (1), Overspeed (2). In each mode, the pump may be faultless (states S, W_1, W_2) or faulty (states F_0, F_1, F_2) and can be repaired only when stopped. The failure rate in the mode 2 is greater than in the mode 1 ($\lambda_2 > \lambda_1$), which is realistic, and there is no failure on demand.

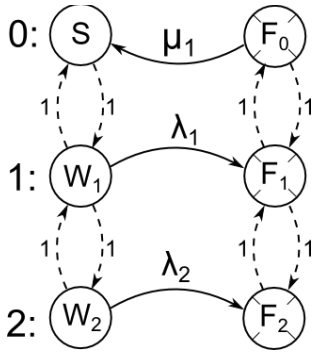


Figure 7: SMP model of a pump (P)

The SMP MV that describes the behavior of a manual valve comprises two modes: Closed (0) and Open (1). It is assumed that no failure can happen because the valves are correctly monitored and periodically tested by well-trained operators.

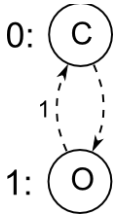


Figure 8: SMP model of a manual valve (MV)

The generic model of a heat exchanger HE is composed of two operation modes: Empty (0) and Full (1). In each mode, the exchanger may be fully operational (states OK_E, OK_F), partially clogged (states PC_E, PC_F) or faulty, i.e. completely blocked (states KO_E, KO_F); it can be repaired only when empty.

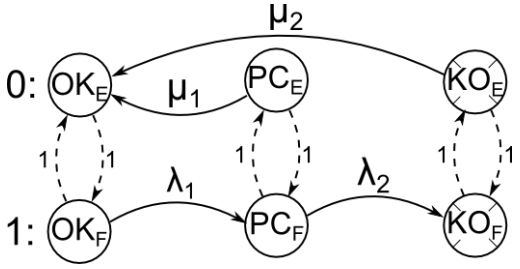


Figure 9: SMP model of a heat exchanger (HE)

A component of the control architecture (PLC or BUS) is merely modeled as a simple component with two states: faultless (W) and faulty (F). As there is only one operation mode for this component (it is assumed never stopped), this mode is numbered 1. This SMP is called CU (Control Unit) in the model.

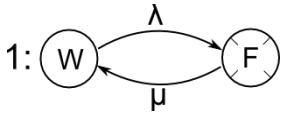


Figure 10: SMP model of a Control Unit (CU)

The behavior of the Primary Circuit Cooling system, not detailed in this paper, is simply modeled by a 2-SMP with two modes Stopped (0) and Run (1),. In each mode, this system may be faultless (states S and W_1) or faulty (states F_0, F_1); it can be repaired only when stopped. This SMP is called SF (Simple Failure) in the model.

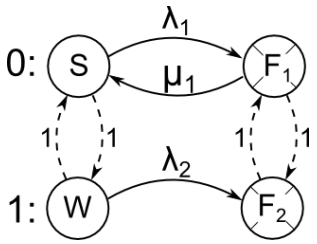


Figure 11: SMP model of a simple failure unit (SF)

Last, the supervision of the plant is described by the SMP SUP which comprises four states that correspond to the different phases of a mission.

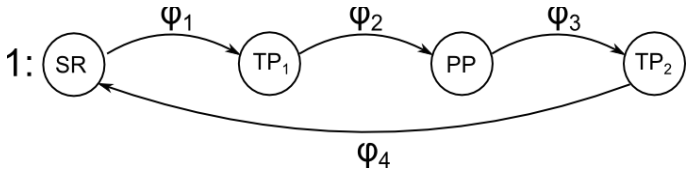


Figure 12: SMP model of the supervision (SUP)

5.2.3 Models of the reconfiguration strategies

The five reconfiguration strategies that have been defined at 5.1 are formally represented at Figure 13 by Moore machines as follows:

- The strategies RS1 and RS2 that require some functions according to the mission phase are respectively formalized by the machines M_{req1} and M_{req2} which are respectively associated to the switches S1 and S2 of the model of Figure 6a.
- The strategy RS3 which defines the replacement of the Primary Circuit Cooling system by the Pools Cooling system is represented by the machine M_{repl} associated to S3.
- Last, the strategies RS4 and RS5 that define the operation modes of the redundant pumps P1 and P2 are given by the machine M_{redu} whose instances are associated to S4 and S5.

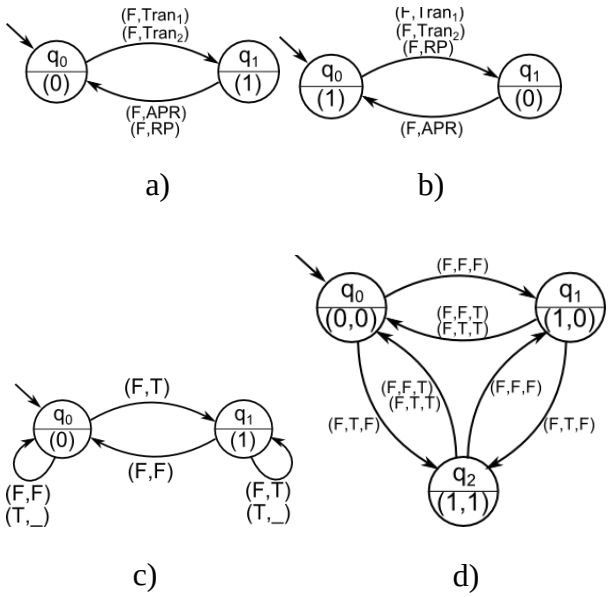


Figure 13: Moore machines models of reconfiguration strategies (a) M_{req1}; b) M_{req2}; c) M_{repl}; d) M_{redu}

5.3 Minimal Cut Sequences

Assuming that the system is initially in the SR phase, the application of Algorithm 1 to the GBDMP which has been constructed at the previous section provides the set of minimal cut sequences $\mathcal{L}_1 = \mathcal{L}_{MCS}(\Subset)$. The shortest elements of this language, sequences whose length is equal to 2, 3, 4 and 5, are reported[6] on Table 1. These sequences are provided in this order by Algorithm 1 and are the most relevant ones for qualitative safety analysis.

Total number of MCS	Selection of MCS
Length 2: 2	$f_a^{P1} f_s^{P2}$ $f_a^{P2} f_s^{P1}$
Length 3: 5	$f_a^{BUS1} f_a^{BUS2} f_a^{P1}$ $f_a^{BUS2} f_a^{BUS1} f_a^{P1}$ $f_a^{PLC2} f_a^{PLC5} f_a^{P1}$ $f_a^{PLC5} f_a^{PLC2} f_a^{P1}$ $\boldsymbol{f_a^{P2} n^{SUP}_{(SR \rightarrow TP1)} f_a^{P1}}$
Length 4: 12	$f_a^{PLC1} f_a^{PLC4} f_a^{P2} f_a^{P1}$ $f_a^{PLC4} f_a^{PLC1} f_a^{P2} f_a^{P1}$ $\boldsymbol{f_a^{P1} f_a^{PLC1} f_a^{PLC4} r_p^{P1}}$ $f_a^{P1} f_a^{PLC4} f_a^{PLC1} r_p^{P1}$ $f_a^{PLC1} f_a^{P1} f_a^{PLC4} r_p^{P1}$ $f_a^{PLC4} f_a^{P1} f_a^{PLC1} r_p^{P1}$ $n_a^{HE1} n_a^{HE2} f_a^{HE2} f_a^{HE1}$

[6] *f, r* and *n* mean *failure*, *repairs* and *neutral* events; *a, s* and *p* mean *active, overspeed* and *passive* mode. Then f_p^{P1} represents the event: *failure of the leaf P1 while it is passive (mode 0)*.

	$n_a^{HE2}\, n_a^{HE1}\, f_a^{HE2}\, f_a^{HE1}$ $n_a^{HE1}\, n_a^{HE2}\, f_a^{HE1}\, f_a^{HE2}$ $n_a^{HE2}\, n_a^{HE1}\, f_a^{HE1}\, f_a^{HE2}$ $n_a^{HE1}\, f_a^{HE1} n_a^{HE2}\, f_a^{HE2}$ $n_a^{HE2}\, f_a^{HE2}\, n_a^{HE1}\, f_a^{HE1}$
Length 5: 8	$\boldsymbol{f_a^{P1}\, f_a^{BUS1}\, f_a^{BUS2}\, r_p^{P1}\, f_s^{P2}}$ $f_a^{P1}\, f_a^{BUS2}\, f_a^{BUS1}\, r_p^{P1}\, f_s^{P2}$ $f_a^{BUS1}\, f_a^{P1}\, f_a^{BUS2}\, r_p^{P1}\, f_s^{P2}$ $f_a^{BUS2}\, f_a^{P1}\, f_a^{BUS1}\, r_p^{P1}\, f_s^{P2}$ $f_a^{P2}\, f_a^{BUS1}\, f_a^{BUS2}\, r_p^{P1}\, f_s^{P1}$ $f_a^{P2}\, f_a^{BUS2}\, f_a^{BUS1}\, r_p^{P1}\, f_s^{P1}$ $f_a^{BUS1}\, f_a^{P2}\, f_a^{BUS2}\, r_p^{P1}\, f_s^{P1}$ $f_a^{BUS2}\, f_a^{P2}\, f_a^{BUS1}\, r_p^{P1}\, f_s^{P1}$

Table 1. Extract of $\mathcal{L}_{MCS}(\Subset)$ for the pools cooling system

The meaning of some of these sequences is easily obtained; when both pumps have failed, the first one in active mode and the second one in the overspeed mode, the system fails obviously, as formally stated by the first two MCS. The interpretation of other MCS is not so straightforward, however. Some a priori puzzling results (in bold characters in the table) are discussed in what follows.

The 3-events long sequence $\boldsymbol{f_a^{P2}\, n^{SUP}{}_{(SR\rightarrow TP1)}\, f_a^{P1}}$ includes a neutral event (phase change from SR to TP_1) and leads nevertheless to a global failure because:

- When the pump P2 fails in the active mode ($\boldsymbol{f_a^{P2}}$), it is deactivated and the pump P1 is switched in the Overspeed mode, according to respectively RS5 and RS4.
- Then, the phase is changed from SR to TP_1. As the RPC function is no more required, P1 is switched in the On mode.
- A global failure occurs when this pump fails in this mode.

The 4-events long sequence $\boldsymbol{f_a^{P1}\, f_a^{PLC1}\, f_a^{PLC4}\, r_p^{P1}}$ is really surprising because it ends with a repair event but leads to a failure. This is nevertheless a valid cut sequence and is explained as follows:

- When P1 fails in the active mode ($\boldsymbol{f_a^{P1}}$), this pump is deactivated and P2 is switched in the Overspeed mode, according to respectively RS4 and RS5.
- Once PLC1 and PLC4 have both failed, RS4 is no more possible.
- Hence, when P1 is repaired in passive mode ($\boldsymbol{r_p^{P1}}$), it cannot be activated (RS4 cannot be applied) but P2 is switched in the On mode, according to RS5. Only the RPC function is performed by this pump; the FPC function is lost.

The reasoning is similar for the 5-events long sequence $\boldsymbol{f_a^{P1}\, f_a^{BUS1}\, f_a^{BUS2}\, r_p^{P1}\, f_s^{P2}}$:

- Once the two buses have failed, no reconfiguration strategy can be applied.
- Hence, when P1 is repaired, P1 remains deactivated and P2 in the Overspeed mode.
- Consequently, a global failure happens when P2 fails because P1 cannot replace it.

It must be underlined that this cut sequence is a MCS even if it includes the sequence $f_a^{P1}\ f_s^{P2}$ because the covering cut of the first sequence ({BUS1, BUS2, P2}) does not includes the covering cut of the second one ({P1, P2}); therefore, the relation $\Subset$ is not satisfied for these two sequences.

5.4 Discussion

This section has shown, on the basis of a real critical system, the relevance of our approach. The MCS that have been obtained, and in particular those which were commented above, cannot be yielded neither from the definition based on sequence inclusion (section 2.2), which does not consider repair events, nor from the definition based on the coherence relation (section 2.3). Safety analysis with the minimality criterion we introduced in section 3.2 is therefore more complete.

VI - 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 because MCS that cannot be found by other approaches are discovered.

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. Development of this approach is a motivating and challenging prospect for future work.

Last, our contribution may be extended to cases where safety analysis is not based on GBDMP models but on GSPN (Generalized Stochastic Petri Nets) or SAN (Stochastic Activity Networks). As our approach to obtain the MCS is based on the analysis of a finite automaton that represents the behavior of the safety analysis model, it should be possible to extend our work by using the automaton equivalent to the selected GSPN or SAN. This is another perspective for further research.

REFERENCES

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

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

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

[Chaux 2013]: Chaux, P.-Y., Roussel, J.-M., Lesage, J.-J., Deleuze, G. & Bouissou M. *Towards a unified definition of minimal cut sequences*. IFAC Proceedings Volumes, 46 (22), 2013, pp. 1-6.

[Ge 2014]: Ge D., Zhang R., Chou Q., Yang Y. *Probabilistic model–based multi-integration formulas for quantifying a generalized minimal cut sequence*. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, vol. 229 (1), pp. 73-82, 2014.

[Kriaa 2012]: Kriaa S., Bouissou M. & Pietre-Cambacedes L. *Modeling the Stuxnet attack with BDMP: Towards more formal risk assessments*. Proceedings of the 7th International Conference on Risks and Security of Internet and Systems (CRiSIS), Cork (Ireland), 2012.

[Li 2015]: Li Y.F. et al. *Dynamic fault tree analysis based on continuous-time Bayesian networks under fuzzy numbers*. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, 229 (6), pp. 530-541, 2015.

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

[Pietre-Cambacedes 2011]: Pietre-Cambacedes L., Deflesselle Y. & Bouissou M. *Security Modeling with BDMP: From Theory to Implementation*. Proc. 5th IEEE Conference on Network and Information Systems Security, Milano (I), pp. 1-8, 2011.

[Piriou 2014]: Piriou, P.-Y., Faure, J.-M. & Lesage J.-J. *Control-in-the-loop Model Based Safety Analysis*. Proc. 23th European Safety & Reliability Conference (ESREL'14). 2014, Wroclaw (Poland).

[Piriou 2017]: 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*. Reliability Engineering & System Safety, 163, pp. 57-68.

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

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

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

Appendix 1: Table of symbols

f_i	Failure event of component i
r_i	Repair event of component i
Σ	Set of events, called alphabet
σ	Sequence of event occurrences
ε	The zero-length sequence
Σ^*	Set of all sequences of event occurrences which can be built on Σ
$\mathcal{L} \subseteq \Sigma^*$	A set of event occurrences sequences, called language
q_i	State i of an automaton
q_0	Initial state of an automaton
Q	Set of states of an automaton
$Q_M \subseteq Q$	Set of the marked states
$\mathcal{L}(\mathcal{A}) = \mathcal{L}_E \subseteq \Sigma^*$	Evolution language of the automaton $\mathcal{A}$
$\mathcal{L}_m(\mathcal{A}) = \mathcal{L}_M \subseteq \mathcal{L}_E$	Marked language of the automaton $\mathcal{A}$
$\mathcal{L}_\mathcal{F} = \mathcal{L}_M$	Failure language for safety analysis
$\mathcal{L}_{CS} \subseteq \mathcal{L}_F$	Set of cut sequences
$\mathcal{L}_{MCS}(\mathcal{R}) \subseteq \mathcal{L}_{CS}$	Set of Minimal Cut Sequences according to an order relation $\mathcal{R}$,

Appendix 2: Detailed description of Algorithm 1

Algorithm 1 – Calculus of the set of MCS

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

1: // *Initialization*
2: $\mathcal{L}_{MCS} := \emptyset$
3: $seqOfLastLength := \{\varepsilon\}$
4: $seqOfCurLength := \emptyset$
5: // *Main loop*
6: **while** $seqOfLastLength \neq \emptyset$ **do**
7: **for all** $\sigma_{last} \in SeqOfLastLength$ **do**
8: $q_{last} := (q_0 , \sigma_{last})$
9: **for all** $u \in (q_{last})$ **do**
10: $q_{cur} := (q_{last} , u)$
11: $\sigma_{cur} := \sigma_{last} u$
12: **if** $q_{cur} \notin Q_M \wedge \nexists \sigma \in Pref(\sigma_{last}) \mid (q_0 , \sigma) = q_{cur}$ **then**
13: $seqOfCurLength := seqOfLastLength \cup \{ \sigma_{cur} \}$
14: **else if** $q_{cur} \in Q_M$ **then**
15: **if** $\nexists \sigma_{min} \in \mathcal{L}_{MCS} \mid \sigma_{min} \Subset \sigma_{cur}$ **then**
16: $\mathcal{L}_{MCS} := \mathcal{L}_{MCS} \cup \{ cur \}$
17: **end if**
18: **end if**
19: **end for**
20: **end for**
21: $seqOfLastLength := seqOfCurLength$
22: $seqOfCurLength := \emptyset$
23: **end while**

Appendix 3: Evolutions of a Moore machine associated to a switch

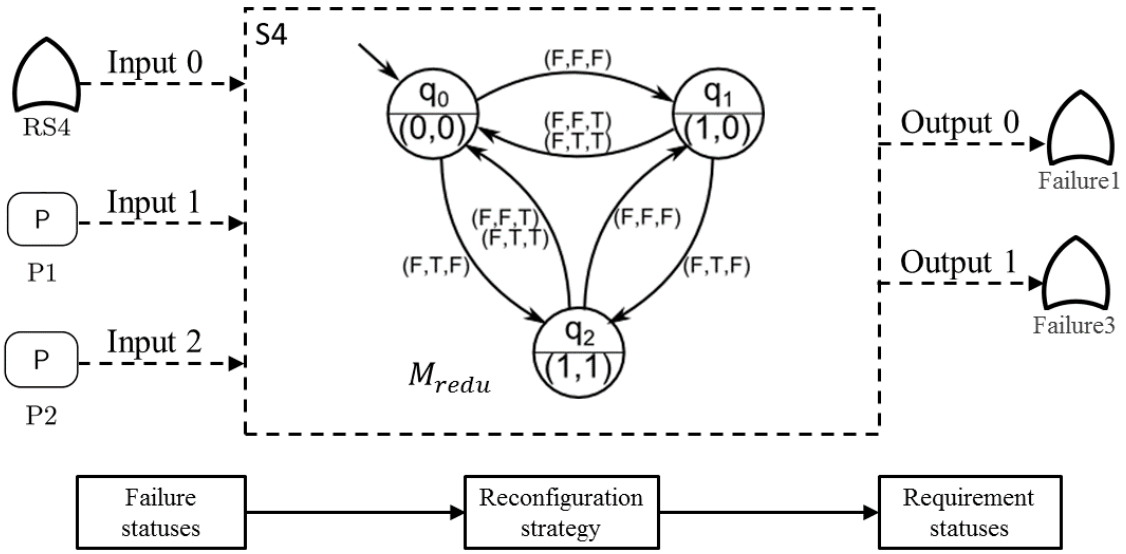


Figure 14: Behavior of the switch S4

The behavior of the switch S4, at Figure 6a, is described by the Moore machine M_{redu} that is represented on Figure 14. The three inputs of this machine are the failure statuses (F(false) or T(true)) of, in this order, the implementation of the reconfiguration strategy on control devices (gate RS4) and of the leaves P1 and P2. The reconfiguration strategy is formally described by the Moore machine M_{redu}. The two outputs are the requirement statuses of the gates Failure1 and Failure3.

The three states of the machine are related to the operation modes of P1 as follows. In the state q_0, where both failure gates are not required, this pump is in its operation mode *Stopped* and in the other two states q_1 and q_2 in its operation modes *On* and *Over* respectively. The transitions between these three states are controlled by the input failure statuses. For example, the Moore machine M_{redu} moves from state q_0 to state q_1 if the three input failure statuses are (F,F,F) (i.e. neither RS4 nor P1 nor P2 is faulty).