Formalisation des scénarios de défaillance d'un BDMP par automate ni

Pierre-Yves Chaux, Jean-Marc Roussel, Jean-Jacques Lesage

Formalisation des scénarios de défaillance d'un BDMP par automate fini

Pierre-Yves CHAUX[1,2], Jean-Marc ROUSSEL [1], Jean-Jacques LESAGE[1]

[1] Laboratoire Universitaire de Recherche en Production Automatisée (LURPA)
61, av. du président Wilson 94235 Cachan cedex, France

[2] EDF R&D - Département MRI
1, av. du Général de Gaulle 92140 Clamart, France

[1] Prénom.Nom@lurpa.ens-cachan.fr,[2] Prénom.Nom@edf.fr

Résumé— **Les Boolean Driven Markov Processes (BDMP) ont été conçus par EDF pour conduire des analyses prédictives de sûreté de systèmes critiques. Tout comme pour les arbres de défaillance, la construction d'un BDMP consiste à décrire les combinaisons de panne des composants d'un système qui conduisent à sa défaillance. Une fois ce modèle établi, la recherche de toutes les séquences de pannes élémentaires qui conduisent à la défaillance globale du système est un problème complexe qui nécessite d'explorer l'espace d'état, le plus souvent de très grande taille, qui est sous-jacent au BDMP. Pour déterminer de manière systématique et exhaustive ces scénarios de défaillance, nous utilisons une approche basée sur la théorie des langages et des automates finis. Nous proposons dans ce papier un algorithme qui permet de calculer un automate fini "équivalent" à un BDMP, dans le sens où l'ensemble des scénarios de pannes et réparations du BDMP coïncide avec les mots du langage généré par l'automate.**

Mots-clés— **Systèmes à Évènements Discrets, Sûreté de fonctionnement, Automates Finis, BDMP, Analyses de sûreté**

I. INTRODUCTION

L'étude prédictive de la sûreté des systèmes critiques fait l'objet de nombreux travaux de recherche visant le plus souvent deux objectifs complémentaires [1]. La conduite *d'analyses qualitatives* sur un modèle de sûreté permet de déterminer quels sont les scénarios de panne des composants d'un système (ou plus rarement des scénarios incluant pannes et réparations des composants) qui conduisent à la défaillance globale de l'installation. *L'analyse quantitative* permet de calculer la probabilité de défaillance globale du système étudié ou de défaillance de certains sous-systèmes.

Les Boolean Driven Markov Processes (BDMP) résultent du savoir faire d'EDF en matière de modélisation et d'analyse de la sûreté des systèmes complexes. Il s'agit d'un modèle de la famille des arbres de défaillance [2] qui en étend certaines possibilités de modélisation. Les BDMP permettent en effet de modéliser non seulement les pannes mais également les réparations des composants constituant un système et offrent par ailleurs des possibilités étendues de modélisation de mécanismes de redondance, non seulement de composants élémentaires mais également de sous-systèmes complets.

Nos travaux ont pour objectif l'analyse qualitative des BDMP. Nous ne les manipulerons donc qu'en privilégiant ce seul point de vue, faisant ainsi volontairement abstraction de ses capacités à modéliser les systèmes stochastiques et ses apports pour l'analyse quantitative de la sûreté de fonctionnement. L'analyse qualitative est basée sur le dénombrement de la combinatoire de l'occurrence des événements de panne et de réparation qui est sous-jacente au BDMP. Pour expliciter cette combinatoire, nous proposons d'utiliser la théorie des langages et des automates finis (AF). Pour ce faire, nous traduisons chaque arrangement de panne(s) et réparation(s) de composant(s) élémentaire(s) en une séquence d'événements (ou "mot"), dont l'ensemble constitue un langage. Ce langage étant régulier, nous proposons un algorithme qui permet de construire l'AF qui le génère. L'analyse des défaillances, qui ne peut être conduite directement sur le BDMP, est menée sur l'automate fini "équivalent" qui représente explicitement ces scénarios. Cette représentation formelle permet en effet l'utilisation de nombreux résultats, publiés sur leur manipulation, notamment dans le cas des espaces d'état de grandes tailles.

La suite de ce papier est organisée de la manière suivante : l'utilisation d'un exemple basique nous permet dans un premier temps de décrire les possibilités de modélisation offertes par les BDMP. Nous montrons ensuite dans la partie III comment traduire tous les scénarios de défaillance et de réparation, implicitement décrits par un BDMP, en séquences d'événements constituant un langage. Le processus de génération d'un automate fini générant ce langage, et donc de ce point de vue équivalent au BDMP, est alors explicité dans la section IV.

II. EXEMPLE INTRODUCTIF, MODÉLISATION PAR BDMP

Afin d'illustrer notre approche et de décrire les possibilités qu'offrent les BDMP pour la modélisation des scénarios de défaillance, nous avons retenu l'exemple basique du système de pompage représenté sur la figure 1.

Ce système a pour fonction principale l'alimentation en fluide d'une sortie à l'aide de deux lignes de pompage, composées chacune d'une vanne et d'une pompe.

La modélisation des scénarios de défaillance de ce système comporte plusieurs difficultés classiquement rencon-

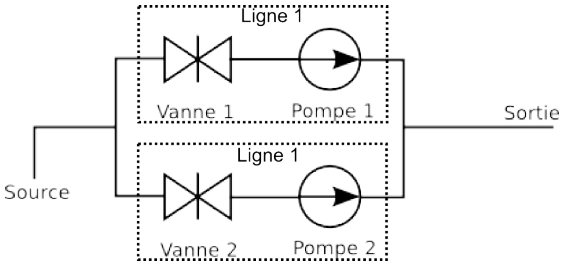

Fig. 1. Système de pompage composé de deux lignes en redondance passive

trées dans les systèmes plus complexes :

- les deux lignes de pompage sont en redondance passive. Le fonctionnement normal est assuré par la seule ligne {Pompe 1 + Vanne 1}. En cas de panne de celle-ci la seconde ligne est mise en marche. Une fois la ligne 1 de nouveau opérationnelle, celle-ci est à nouveau utilisée ;
- tous les composants modélisés (vannes et pompes) sont réparables ;
- alors que les vannes ne peuvent défaillir que lorsqu'elles sont en marche, les pompes peuvent défaillir lorsqu'elles sont en marche (par exemple suite à la perte de l'alimentation électrique) ou bien lorsqu'elles sont à l'arrêt (suite par exemple à un grippage des parties tournantes).

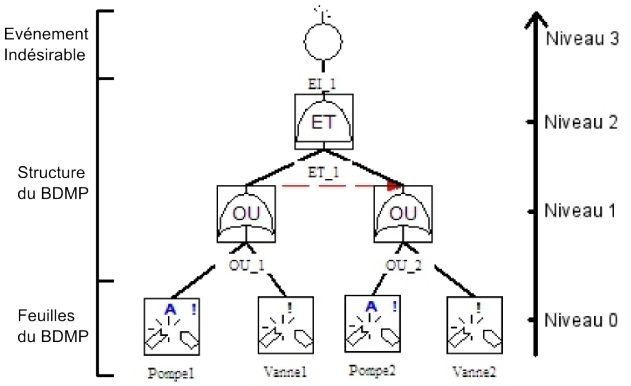

Fig. 2. BDMP modélisant les scénarios de défaillance du système de pompage

Les Boolean Driven Markov Processes (BDMP) ont été proposés par M. Bouissou pour offrir à l'ingénieur des primitives simples lui permettant d'exprimer de tels mécanismes complexes de défaillance [3].
Construits sur la base des arbres de défaillance [2], les BDMP intègrent les possibilités de modélisation des arbres de défaillance dynamiques (ADD), tels que la modélisation des redondances de composants par des portes "Spares" proposés par J.-B. Dugan [4], mais permettent de plus :

- de modéliser des mécanismes de *redondance entre sous-systèmes* (et pas uniquement entre composants) ;
- de représenter de manière native les réparations de composants. L'utilisation des ADD nécessite en effet de faire l'hypothèse que les composants sont non réparables, même si l'ajout de "Repair Boxes" aux ADD proposé par A. Bobbio permet de lever partiellement cette hypothèse [5].

Ainsi, le BDMP de la figure 2 modélise la défaillance du système de pompage de la manière suivante :

- l'événement indésirable (EI), placé à la tête de l'arbre représente la défaillance globale du système (la fonction principale de distribution de fluide n'est plus assurée) ;
- les éléments de bases (les feuilles de l'arbre) représentant les composants élémentaires (une feuille par composant). Chaque type de feuille correspond à un mécanismes de défaillance et de réparation ;
- la structure d'arbre de défaillance décrit, grâce à l'utilisation de portes logiques de type ET et OU, les combinaisons de défaillance des composants élémentaires qui conduisent à la défaillance du système ;
- la modélisation du mécanisme de redondance (entre les deux lignes de pompage) est décrit à l'aide de Gâchettes (représentées en traits pointillés). Cette primitive permet de relier la défaillance du sous-système principal (la ligne 1) par la sortie de la porte OU_1 qui est l'origine de la gâchette, à la sollicitation du sous-système secondaire (la ligne 2) situé à l'extrémité de la gâchette. A la réparation de la ligne 1, la ligne 2 repasse en mode non sollicité afin de retrouver la configuration initiale où la fonction de distribution est assurée par la ligne 1.

Ainsi, la sortie de fluide n'est plus alimentée si la ligne de pompage 1 ET si la ligne de pompage 2 sont défaillantes. Chacune des lignes est défaillante si la pompe OU la vanne est défaillante. Les pompes pouvant défaillir alors qu'elles sont actives ou dormantes, sont associées à des feuilles de type "AF". Les vannes, qui ne peuvent défaillir que lorsqu'elles sont actives, sont quant à elles associées à des feuilles de type "F". Chacun de ces composants peut être réparé quelque soit le mode dans lequel il se trouve (actif ou dormant).

III. Analyse sémantique d'un BDMP

L'analyse qualitative d'un BDMP est pour une large part similaire à l'analyse qualitative des ADD. Il s'agit dans les deux cas d'extraire d'un modèle qui n'exprime pas explicitement de relation séquentielle entre l'apparition des événements de panne (et également de réparation dans le cas des BDMP) celles de ces séquences, appelées *séquences de coupe* [6], qui conduisent à la défaillance globale du système. Les différentes approches permettant l'analyse des ADD que l'on peut trouver dans la littérature proposent des techniques de traduction dans différents formalismes, plus ou moins bien adaptés à cette analyse qualitative.
Les traductions d'un ADD en chaînes de Makov (JB. Dugan [4] et D. Coppit [7]), réseaux bayésiens à temps continu (H. Boudali [8]), réseaux de Petri stochastiques (A. Bobbio [5]), ou plus récemment en Interactive Markov Chains (H. Boudali [9]), privilégient l'objectif de réaliser des analyses quantitatives. Parmi ces différentes approches, seule la traduction en RdP nous paraît pouvoir être utilisée efficacement pour l'analyse qualitative, mais la classe de RdP utilisée (des "Well Formed Stochastic Petri Net") rend difficile la génération du graphe des marquages accessibles, porteur des séquences de coupe.
Plus spécifiquement dédiée à l'analyse qualitative, la démarche proposée par Z. Tang [6] utilise des Zero-suppressed

Binary Decision Diagram après avoir remplacé les portes dynamiques de l'ADD par des portes statiques "équivalentes" et traduit les relations causales entre événements de panne d'un ADD. Les contraintes de séquentialité entre événements, qui traduisent la nature dynamique des ADD, sont ensuite retranscrites de manière heuristique dans le modèle précédemment obtenu. Cette approche, qui n'est pas complètement automatique, conduit à calculer un sur-ensemble des séquences de coupe et n'est donc pas une méthode exacte, même si elle est conservative. Une approche purement algébrique, visant à la fois l'analyse qualitative et l'analyse quantitative des ADD a été récemment proposée par G. Merle [10]. Grâce à ces travaux, la fonction de structure de tout ADD peut être déterminée analytiquement sous une forme canonique et l'ensemble des séquences de coupe en être déduit automatiquement. Cependant, comme pour l'ensemble des travaux cités plus haut, l'hypothèse de non réparabilité des composants d'un ADD est retenue. Un BDMP permettant de modéliser les mécanismes de panne et de réparation des composants, il est donc nécessaire de construire une technique de traduction spécifiquement adaptée. Pour ce faire, nous allons maintenant examiner les scénarios de panne/réparation de composants élémentaires décrits par un BDMP sous l'angle de séquences d'événements constituant un langage pouvant être représenté par un AF.

A. Des scénarios de panne/réparation à l'AF

L'ensemble de tous les scénarios possibles de panne et de réparation de composants élémentaires modélisés par un BDMP peut être représenté par un *langage* $\mathcal{L}$ composé de mots sur un *alphabet* Σ. Cet alphabet Σ est l'union des alphabets Σ_f et Σ_r, respectivement alphabet des événements de pannes et alphabet des événements de réparation. $\mathcal{L}$ comporte donc toutes les séquences d'événements de panne ou de réparation décrits par un BDMP, dont un sous-ensemble ne compromet pas le fonctionnement normal (la fonction principale est assurée), et dont un autre sous-ensemble conduit à la défaillance globale du système (la fonction principale n'est plus assurée).
Ce langage $\mathcal{L}$ est préfixe-clos (tous les préfixes de tous les mots de $\mathcal{L}$ sont également des mots de $\mathcal{L}$), car durant la vie du système chaque nouveau mot de $\mathcal{L}$ est obtenu par concaténation d'un événement élémentaire de panne ou de réparation d'un composant et d'un mot de $\mathcal{L}$ qui traduit son passé. $\mathcal{L}$ est par ailleurs un langage régulier est peut donc être représenté par un AF qui le génère.
La traduction de toutes les séquences d'événements de panne/ réparation d'un BDMP dans un AF permet donc de pratiquer les analyses qualitatives indispensables aux études prédictives de sûreté. Nous allons maintenant expliciter comment extraire les scénarios de panne/réparation d'un BDMP et les traduire en séquences d'événements sur Σ.

B. Modélisation événementielle des feuilles du BDMP

Chaque composant élémentaire du système étudié est associé à une feuille du BDMP. Dans le cadre de cette étude, deux types de feuilles sont retenus :

- les feuilles de type F permettent de décrire des défaillances ne pouvant survenir que lorsque le composant est actif ,
- les feuilles de type AF permettent de décrire des défaillances pouvant survenir lorsque le composant est actif ou dormant .

Dans les deux cas la réparation peut occurrer lorsque le composant est actif ou dormant. L'AF de la figure 3 propose un modèle enveloppe des comportements de ces deux types de feuilles.

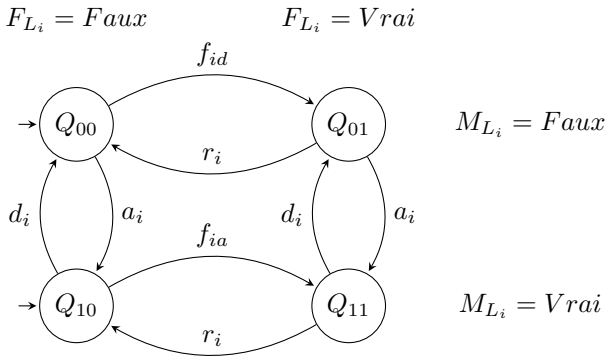

Fig. 3. Enveloppe du comportement d'une feuille L_i d'un BDMP

Quatre états sont utilisés pour représenter qu'un composant est défaillant ou non (variable booléenne F_{L_i}) et selon que le composant est en mode actif ou en mode dormant (variable booléenne de mode M_{L_i}). A chaque instant un seul de ces états est actif. Un composant étant supposé initialement non défaillant, l'état initial de cet AF est Q_{00} ou Q_{10} selon que la feuille est initialement actif ou non. Compte tenu des définitions des deux booléens M_{L_i} et F_{L_i}, les deux états de défaillances d'un composant sont Q_{01} et Q_{11}. les deux état d'activité du composants sont Q_{10} et Q_{11}.

L'AF associé à une feuille i est défini sur l'alphabet $\Sigma_i = \{f_{ia}, f_{id}, r_i, a_i, d_i\}$
où :

- f_{ia} est l'événement de panne du composant i alors qu'il est actif,
- f_{id} est l'événement de panne du composant i alors qu'il est dormant,
- r_i est l'événement de réparation du composant i,
- a_i est l'événement de sollicitation du composant i qui le fait passer du mode dormant au mode actif,
- d_i est l'événement de sollicitation du composant i qui le fait passer du mode actif au mode dormant.

Les BDMP font l'hypothèse de non simultanéité d'occurrence des événements de panne ou de réparation des composants élémentaires. Le comportement d'une feuille AF correspond à l'automate de la figure 3 et le comportement d'une feuille de type F est déduit de cet AF en ôtant l'événement f_{id} de Σ_i et en supprimant la transition (Q_{00}, f_{id}, Q_{01}).

C. Modélisation logique de la structure arborescente d'un BDMP

La structure d'arbre de défaillance d'un BDMP qui est construite "au dessus" des feuilles ne décrit que les combinaisons des défaillances des composants élémentaires qui conduisent à la défaillance globale du système (appelée

événement indésirable - EI). A l'instar des arbres de défaillance statiques, la fonction de structure est l'équation booléenne qui définit ces relations causales. Ainsi pour l'exemple du système de pompage et son BDMP représenté sur la figure 2, la défaillance globale n'est obtenue que si aucune des deux lignes n'est plus à même d'alimenter la sortie. Puisque les booléens F_{L_i}, avec $L_i \in L = \{L_{pompe1}, L_{pompe2}, L_{vanne1}, L_{vanne2}\}$, représentent les états défaillants des 4 composants (cf. AF de la figure 3), la fonction de structure associée au BDMP de la figure 2 est :

$$EI = (F_{L_{pompe1}} \vee F_{L_{vanne1}}) \wedge (F_{L_{pompe2}} \vee F_{L_{vanne2}}) \quad (1)$$

D. Modélisation logique du mécanisme de gâchette

De manière à préciser le mécanisme de gâchette qui a été rapidement décrit dans le §II, nous allons reutiliser le BDMP de la figure 2.

Ce BDMP présente une structure arborescente où trois sous-systèmes (ET_1, OU_1, OU_2) et quatre feuilles ($Pompe_1$, $Pompe_2$, $Vanne_1$, $Vanne_2$) sont interconnectés par un mécanismes de redondance décrit par la structure de gâchette. A chacun de ces trois sous-systèmes est associé une variable de sollicitation M_i qui décrit le fait qu'il soit mis en marche soit à l'initialisation soit par le mécanisme de gâchette (passage en mode actif lorsque le système principal est défaillant). Chacune des feuilles ($L_i \in L$) est associée à une variable de mode M_{L_i} telle qu'elle est décrite dans la section III-B.

La variable logique associée à la sollicitation du sous-système dont le sommet est OU_2 (notée S_{OU_2}) n'est donc mise à 1 que si l'un des sous-systèmes "consommateur" de OU_2 est actif (dans cet exemple, ET_1 uniquement) et si la variable logique associée à la gâchette dont l'origine est OU_1 et la destination OU_2 (notée $G_{OU_1-OU_2}$) est Vraie. Celle-ci passe du niveau logique 0 au niveau logique 1 lorsque le sous-système OU_1 est défaillant, on a alors $G_{OU_1-OU_2} = (F_{L_{pompe1}} \vee F_{L_{vanne1}})$.

De manière générale, si M_i représente la sollicitation du sous-système i et G_{j-i} la variable booléenne représentant la défaillance du sous-système j à l'origine de la gâchette G_{j-i} alors M_i est définie par la fonction booléenne conditionnée par l'existence cette gâchette ou de sous-systèmes consommateurs suivant 4 cas :

- S'il existe au moins un consommateur de i, mais pas de gâchette dont i est la destination :
  $M_i = \bigvee_{\{j/j \in consom(i)\}}(M_j)$
- S'il existe au moins un consommateur de i, et au moins une gâchette dont i est la destination :
  $M_i = \left(\bigvee_{\{j/j \in consom(i)\}}(M_j)\right) \wedge \left(\bigwedge_{\{k\}}(G_{k-i})\right)$
- S'il n'existe pas de consommateur de i, mais au moins une gâchette dont i est la destination :
  $M_i = \bigwedge_{\{k\}}(G_{k-i})$
- S'il n'existe pas de consommateur de i et pas de gâchette dont i est la destination :
  $M_i = False$

En considérant le niveau maximal (ici 3) comme étant celui de l'événement indésirable, toujours sollicité ($S_{EI} = 1$), il est possible de déterminer toutes les sollicitations des sous-systèmes en descendant à travers les niveaux du BDMP. Ainsi les modes des feuilles ($Pompe_1$, $Pompe_2$, $Vanne_1$, $Vanne_2$), toutes placées au niveau 0, peuvent être déterminés par substitution à travers les niveaux du BDMP de manière à décrire le mode d'une feuille M_{L_i} uniquement en fonctions des variables logiques de gâchette G_{j-k}, calculés à partir des F_{L_i}.

IV. Génération d'un AF "équivalent" à un BDMP

La sémantique des BDMP en vue d'une analyse qualitative étant précisée, nous allons maintenant définir une technique pour construire un AF "équivalent" à un BDMP, au sens ou le langage généré par l'AF coïncide avec l'ensemble des scénarios de panne et de réparation implicitement contenus dans un BDMP.

A. Définition du formalisme de description

Un AF est défini par le 5-uplet $< \Sigma, Q, q_0, Q_m, \delta >$ où :
- Σ est l'alphabet d'entrée de l'automate,
- Q est l'ensemble des états de l'automate,
- q_0 est l'état initial tel que $q_0 \in Q$,
- Q_m est l'ensemble des états marqués de l'automate tel que $Q_m \subset Q$,
- δ est la fonction de transition telle que $\delta : (Q, \Sigma) \to Q$

L'alphabet d'entrée Σ est l'ensemble des évènements de défaillance f_{ia}, f_{id} et de réparation r_i de toute les feuilles i, tel que $\Sigma = \Sigma_r \cup \Sigma_f$.
A chaque état q_i de Q est associé un état $\mathbf{F}_{q_i}$ du BDMP, c'est-à-dire une combinaison de l'état défaillant ou non de chacun des composants du BDMP. Le nombre maximum d'états de l'automate est donc borné par 2^n où $n = Card(L)$ est le nombre de feuille du BDMP.
A l'initialisation d'un BDMP, on considère qu'aucune des feuilles n'est défaillante. Cette combinaison est celle associée à q_0.
Nous avons choisi de définir l'ensemble Q_m des états marqués de l'automate comme l'ensemble des états associés à la défaillance globale du système. Ce choix nous permet de calculer l'ensemble des séquences de coupe comme le langage marqué par l'AF équivalent au BDMP.
La fonction de transition δ permet de traduire les conséquences de l'occurrence des événements de défaillance et de réparation sur les changements d'états de l'automate.
Σ et q_0 sont facilement déduits d'un BDMP donné. Dans la suite nous allons donc plus spécifiquement développer comment obtenir Q, Q_m et δ puis proposer un algorithme permettant de construire de manière systématique un AF équivalent à un BDMP.

B. Mécanisme de génération de l'automate "équivalent"

La génération de l'automate traduisant le comportement d'un BDMP consiste pour l'essentiel à calculer l'ensemble Q des états et δ, la fonction de transition. Ce calcul est réalisé de proche en proche à partir de l'état initial (dans lequel touts les composants - ou feuilles - sont en état de fonctionnement) en parcourant l'espace des états du BDMP. Elle est réalisée par l'algorithme 1.

Un état du BDMP est complètement défini par une combinaison des états de défaillance de toutes les feuilles du BDMP. A chaque état q_i du BDMP est donc associé un vecteur d'état $\mathbf{F_{q_i}}$ de dimension $Card(L)$ dont chaque composante est une variable booléenne F_{L_j} représentant la

Algorithm 1 Generate states and transitions of the "equivalent" FSA of a BDMP

Require: $\Sigma = \Sigma_f \bigcup \Sigma_r$: Symbol alphabet ,$L = \{L_i\}$: leaf dictionnary(contain leaf type),$\{M_{Li}\}$ Boolean mode function of leaf L_i, EI Top event boolean function, $\Delta : (S\mathbf{F}, \Sigma) \rightarrow S\mathbf{F}$.

1: # Generating initial state
2: $\mathbf{F_{q0}} \leftarrow [F_{L_i} = False | L_i \in L]$
3: $S\mathbf{F} \leftarrow \mathbf{F_{q0}}$
4: $Q \leftarrow q_0$
5: **for** each q_i in Q **do**
6: # Step 1 : evaluate $\mathbf{M_{qi}}$
7: $\mathbf{M_{qi}} = [M_{L_j} | L_j \in L]$
8: # Step 2 : generate sensible events SE in state q_i.
9: $SE = Sensible\_events(L, \mathbf{F_{qi}}, \mathbf{M_{qi}})$
10: # Step 3 : Generate next state q_n
11: **for** each $u \in SE$ **do**
12: $\mathbf{F_{qn}} = \Delta(\mathbf{F_{qi}}, u)$
13: **if** $\mathbf{F_{qn}} \notin S\mathbf{F}$ **then**
14: $S\mathbf{F} \leftarrow \mathbf{F_{qn}}$
15: $Q \leftarrow q_n$
16: **if** $EI(F_{L_i} \in \mathbf{F}_{q_n}) = True$ **then**
17: $Q_M \leftarrow q_n$
18: **end if**
19: **end if**
20: $\delta \leftarrow < q_i, u, q_n >$
21: **end for**
22: **end for**
23: **return** $< \Sigma, Q, q_0, Q_m, \delta >$

Algorithm 2 Generate $SE(L, \mathbf{F_{qi}}, \mathbf{M_{qi}})$

Require: $L = \{L_i\}$: leaf dictionnary(contain leaf type), $\mathbf{F} = [F_{Li}]$ faillure vector, $\mathbf{M} = [M_{Li}]$ mode vector

1: $SE := \emptyset$
2: **for** each $L_i \in L$ **do**
3: **if** $\overline{F_{Li}} \wedge \overline{M_{Li}}$ **then**
4: **if** $l_i =' AF'$ **then**
5: $SE \leftarrow \{f_{id}\}$
6: **end if**
7: **else if** $F_{Li} \wedge \overline{M_{Li}}$ **then**
8: $SE \leftarrow \{r_i\}$
9: **else if** $\overline{F_{Li}} \wedge M_{Li}$ **then**
10: $SE \leftarrow \{f_{ia}\}$
11: **else if** $F_{Li} \wedge M_{Li}$ **then**
12: $SE \leftarrow \{r_i\}$
13: **end if**
14: **end for**
15: **return** SE

défaillance de la feuille j du BDMP ($1 \leq j \leq Card(L)$). Ces vecteurs sont stockés dans un ensemble $S\mathbf{F}$. Pour déterminer l'évolution de l'état du BDMP sur occurrence d'un événement de panne ou de réparation (événement $u \in \Sigma$) d'une feuille (fonction δ), la fonction Δ, permet de calculer la nouvelle valeur du vecteur d'état après occurrence de u. Cette fonction est une représentation de l'effet d'une évolution d'un des automates de feuilles (panne ou réparation) sur occurrence d'un événement u. Chaque fois que le système atteint un état de défaillance globale ($EI(F_{L_i}) = 1$) cet état est marqué. La défaillance globale du système est donc représentée par l'ensemble Q_m des états marqués de l'AF ($Q_m \subset Q$). Les sollicitations élémentaires M_{L_i} associée à chacune des feuilles L_i sont mises à jour. Les modes des feuilles sont ensuite placées dans le vecteur $\mathbf{M_{qi}}$ associé à l'état courant q_i.

Après la génération de l'état initial q_0 associé à l'ensemble $\mathbf{F_{q0}}$ des booléens de défaillance tous faux à l'état initial (lignes 1-4), l'algorithme lance la calcul des états suivants à partir de l'état courant $q_i \in Q$ grâce à la boucle «FOR» située entre les lignes 5 à 22. La première étape consiste à calculer M_{q_i}, le vecteur de sollicitations des feuilles à l'état considéré (ligne 7). Une fois $\mathbf{F_{qi}}$ et $\mathbf{M_{qi}}$ connus, on déduit dans quel état se trouvent chacune des feuilles. La ligne 9 fait ensuite appel au second algorithme qui calcule SE, le sous-ensemble de Σ auquel le BDMP est sensible dans l'état courant q_i. L'algorithme 2 génère donc SE en fonction de l'état actif (connu grâce à $\mathbf{F_{qi}}$ et $\mathbf{M_{qi}}$) de chacune des feuilles et du type de feuille, stocké dans l'ensemble L. Une fois l'ensemble SE connu la boucle «FOR» située entre les lignes 11 et 21 calcule pour chaque couple $(q_i, u \in SE)$ l'état suivant q_n. La ligne 14 permet de calculer quel est l'effet de l'événement u sur le vecteur $\mathbf{F_{qi}}$ des niveaux de défaillance des feuilles du BDMP pour obtenir sa nouvelle valeur $\mathbf{F_{qn}}$. Afin de savoir si ce nouvel état est déjà dans Q, $\mathbf{F_{qn}}$ est comparé aux vecteurs stockées dans $S\mathbf{F}$ (ligne 13). Si ce vecteur est nouveau alors q_n est placé dans Q (ligne 15) et $\mathbf{F_{qn}}$ dans $S\mathbf{F}$ (ligne 114). Si la situation $\mathbf{F_{qn}}$ est une situation de défaillance globale ($EI(F_{L_i} = 1)$ alors q_n est également placé dans Q_M(lignes 16-18). Que le nouvel état soit déjà connu ou nouveau, il est nécessaire de placer la transition (q_i, u, q_n) dans la fonction de transition δ (ligne 20). Une fois l'algorithme terminé on retourne le 5-uplet décrivant l'automate fini (ligne 23). La convergence de l'algorithme 1 est assurée par le nombre d'états que l'on est susceptible de calculer, borné à $2^{Card(L)}$.

C. Automate "équivalent" au BDMP de la figure 2

L'ensemble $L = L_i$ des feuilles du BDMP du système de pompage est : $L = \{L_{pompe1}, L_{vanne1}, L_{pompe2}, L_{vanne2}\}$. A chaque feuille doit également être associé son type (type F lorsque les défaillances ne peuvent survenir que lorsque le composant est en marche, type AF lorsque les défaillances peuvent survenir lorsque le composant est en marche ou à l'arrêt) :

Feuille	Type	Événements associées
L_{pompe1}	AF	f_{0d}, f_{0a}, r_0
L_{vanne1}	F	f_{1a}, r_1
L_{pompe2}	AF	f_{2d}, f_{2a}, r_2
L_{vanne2}	F	f_{3a}, r_3

La fonction de structure de la partie statique du BDMP est donnée par (cf. §III.C) :
$EI = (F_{L_{pompe1}} \vee F_{L_{vanne1}}) \wedge (F_{L_{pompe2}} \vee F_{L_{vanne2}})$
Enfin, la sollicitation des feuilles, calculée à partir des fonctions booléennes de sollicitation (cf. §III.D) est :

- $M_{L_{pompe1}} = 1$
- $M_{L_{vanne1}} = 1$
- $M_{L_{pompe2}} = F_{L_{pompe1}} \vee F_{L_{vanne1}}$
- $M_{L_{vanne2}} = F_{L_{pompe1}} \vee F_{L_{vanne1}}$

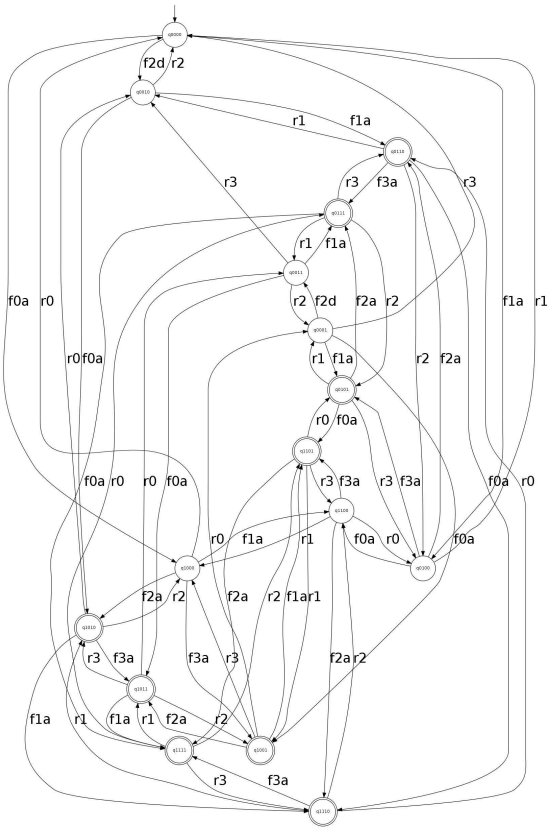

Fig. 4. Comportement du BDMP représentant le système de pompes (AF graphique obtenu à l'aide de Graphviz)

A partir de ces données automatiquement extraites du BDMP, l'algorithme 1 génère l'automate représenté fig. 4. Cet automate est composé de 16 états. Toutes les combinaisons possibles de défaillance des composants sont donc atteignables à partir de l'état initial. Parmi ces 16 états, 9 sont marqués, ils représentent à eux 9 la défaillance globale du système de pompage.

Une vérification formelle de la cohérence sémantique entre le BDMP de la figure 2 et l'AF de la figure 4 n'est pas possible. Cependant, compte tenu de la simplicité du cas d'étude, il est aisé de retrouver dans cet AF quelques caractéristiques des mécanismes de panne/réparation que le concepteur à transcrit dans le BDMP. Ainsi par exemple, trois transitions étiquetées par des événements de défaillance sortent de l'état initial. Elles correspondent aux défaillances de la ligne 1 en mode actif (panne possible de la pompe 1 et de la vanne 1) ainsi qu'à la défaillance de la ligne 2 en mode dormant (panne de la pompe 2 seulement car la vanne 2 ne peut tomber en panne en mode dormant). Il est également possible de remarquer en parcourant l'automate que la défaillance de la vanne 2 ne peut intervenir qu'après défaillance de la pompe 1 ou de la vanne 1 (ce qui n'est pas le cas pour la pompe 2). Enfin, pour tout composant, l'événement de réparation n'occurre qu'après l'événement de panne.

V. Conclusions et perspectives

Dans ce papier, nous avons décrit une technique permettant de générer automatiquement un automate à états fini équivalent à un BDMP, au sens où le langage généré par l'AF coïncide avec l'ensemble des scénarios de panne et de réparation décrits implicitement par le BDMP. Pour ce faire, une étude approfondie de la sémantique logique et événementielle des BDMP a été nécessaire.

De nombreuses études qualitatives de la sûreté prévisionnelle d'un système peuvent être conduites sur cet AF. Par exemple, l'ensemble des séquences de coupe du BDMP (l'ensemble des séquences d'événements de panne/réparation de composants qui conduisent à la défaillance globale de l'installation) peut être aisément déterminé en calculant le langage marqué par l'AF.

Nos travaux actuels portent d'une part sur la formalisation des feuilles autorisant la défaillance des composants au moment de la sollicitation (type de feuilles que nous n'avons pas traité dans cette publication). D'autre part, l'AF obtenu à partir d'un BDMP décrivant les défaillances d'un système complexe (comportant plusieurs dizaines de feuilles) étant de grande taille, nous testons l'efficacité des techniques de Model-Checking pour la détermination des séquences de coupe. Enfin, la recherche des séquences minimales de coupe [6] revêt une importante particulière en vue des études quantitatives. Ce travail nécessite la détermination de critères de minimalité des séquences de coupe qui n'ont pas encore été formellement définis dans la littérature.

Références

[1] E.J. Henley et H. Kumamoto : *Reliability engineering and risk assessment*. Prentice-Hall Englewood Cliffs (NJ), 1981.

[2] M. Stamatelatos, W. Vesely, J. Dugan, J. Fragola, J. Minarick et J. Railsback : Fault tree handbook with aerospace applications, 2002.

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

[4] JB Dugan, SJ Bavuso et MA Boyd : Dynamic fault-tree models for fault-tolerant computer systems. *IEEE Transactions on reliability*, 41(3):363–377, 1992.

[5] A. Bobbio et DC Raiteri : Parametric fault trees with dynamic gates and repair boxes. *In Reliability and Maintainability, 2004 Annual Symposium-RAMS*, pages 459–465, 2004.

[6] Z. Tang et JB Dugan : Minimal cut set/Sequence generation for dynamic fault trees. *In Reliability and Maintainability, 2004 Annual Symposium-RAMS*, pages 207–213, 2004.

[7] D. Coppit, K.J. Sullivan et J.B. Dugan : Formal semantics of models for computational engineering : a case study on dynamic fault trees. *In 11th International Symposium on Software Reliability Engineering, 2000. ISSRE 2000. Proceedings*, pages 270–282, 2000.

[8] H. Boudali et J.B. Dugan : A new Bayesian network approach to solve dynamic fault trees. *In Reliability and Maintainability Symposium*, volume 17, 2005.

[9] H. Boudali, P. Crouzen et M. Stoelinga : A compositional semantics for Dynamic Fault Trees in terms of Interactive Markov Chains. *Lecture Notes in Computer Science*, 4762:441, 2007.

[10] G. Merle, J.-J. Lesage et J.-M. Roussel : Algebraic Determination of the Structure Function of Dynamic Fault Trees. *Reliability Engineering and System Safety*, 96(2):267–277, 2011.