# Évaluation dynamique d'indicateurs de sûreté de fonctionnement pour des architectures de contrôle-commande par le biais de réseaux de Petri colorés stochastiques

Grâce Boyer, Moulaye Ndiaye, Nicolae Brînzei, Jean-François Pétin, Jacques Camerini

Institut pour la Maîtrise des Risques
Sûreté de Fonctionnement - Management - Cindyniques

# Évaluation dynamique d'indicateurs de sûreté de fonctionnement pour des architectures de contrôle-commande par le biais de réseaux de Petri colorés stochastiques

# Dynamic dependability indicators assessment of industrial control systems through colored stochastic Petri nets

Grâce Boyer
*Université de Lorraine, CNRS,CRAN, F-54000*
Nancy, France
grace.boyer@univ-lorraine.fr

Nicolae Brînzei
*Université de Lorraine, CNRS,CRAN, F-54000*
Nancy, France
nicolae.brinzei@univ-lorraine.fr

Jacques Camerini
*Industrial Automation, Schneider Electric France*
Carros, France
jacques.camerini@se.com

Moulaye Ndiaye
*Industrial Automation, Schneider Electric France*
Carros, France
moulaye.ndiaye@se.com

Jean-François Pétin
*Université de Lorraine, CNRS,CRAN, F-54000*
Nancy, France
jean-francois.petin@univ-lorraine.fr

***Résumé*— En avant-vente d'un projet, une évaluation des performances de sûreté de fonctionnement d'une architecture de contrôle-commande est réalisée grâce à un modèle décrivant son comportement dynamique. Elle permet de déterminer l'impact des composants sur l'ensemble en soulignant les points de défaillance uniques.**

***Summary*— In pre-sales phase, a dependability performances assessment of an industrial control system is performed using a model describing its dynamic behavior. This assessment determines the impact of the components on the global system by highlighting the single points of failure.**

***Mots-clés—Réseau de Petri stochastique, simulation de Monte Carlo, Architecture de contrôle-commande, diagrammes UML, évaluation fiabiliste***

## I. INTRODUCTION

Un procédé industriel est un ensemble de tâches à réaliser pour fabriquer des produits ou pour fournir un service et qui est automatisé. Il se repose sur une architecture de contrôle-commande qui est un ensemble de composants, matériels et logiciels, dédiés à l'automatisation de ce procédé industriel. Sa définition est basée sur un diagramme de flux et d'instrumentation du procédé (P&ID) définissant ses différentes tâches mais également sur les demandes du client spécifiant les performances qu'il souhaite sur l'architecture. Parmi ces performances, il est de plus en plus fréquent dans les projets de retrouver des exigences de la sûreté de fonctionnement. En effet, ces dernières ont un impact non négligeable à la fois sur la gestion des risques du projet et sur le cycle de vie de l'architecture, ainsi que sur la sécurité des opérateurs ou la protection de l'environnement. Par exemple, le contrôle-commande d'une station d'épuration des eaux usées (Fig. 1) doit être en mesure de fournir un service continu du procédé sans interruption pour assurer la qualité du traitement de l'eau.

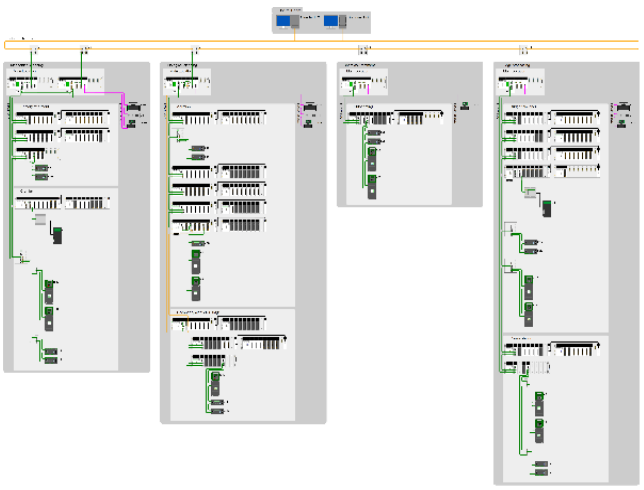

Fig. 1. Exemple d'une architecture de contrôle-commande pour une station d'épuration des eaux usées pour 100 000 habitants.

Pour garantir ces performances de sûreté de fonctionnement, il est nécessaire de les évaluer pendant la phase d'avant-vente. Cependant, cela reste difficile en raison de la faible quantité d'informations sur le projet en avant-vente qui se limitent au P&ID et aux spécifications dans le cahier des charges, mais également par une limite des ressources humaines et financières ainsi que d'un délai restreint pour la soumission d'une proposition d'architecture au client. De plus, le comportement dynamique de l'architecture, couplé au phénomène de vieillissement des composants, constitue une contrainte supplémentaire à prendre en considération.

Le principal objectif de nos travaux est de fournir aux ingénieurs en avant-vente une méthode capable d'évaluer les performances de sûreté de fonctionnement pour tous types d'architecture de contrôle-commande. Cet article présente nos travaux ainsi que les perspectives.

La section 2 présente le défi industriel qui se pose et justifie le choix des réseaux de Pétri (RdP) comme méthode appropriée pour développer des modèles formels d'évaluation des performances de sûreté de fonctionnement. Nos travaux de recherche sont introduits dans la section 3 à travers la description de l'évaluation. La section 4 présente les perspectives en mettant en évidence les différents aspects qui pourraient être ajoutés pour améliorer l'évaluation. L'article se termine par une conclusion dans la section 5.

## II. PROBLÉMATIQUE INDUSTRIELLE

### A. *Exigences pour l'évaluation fiabiliste*

Un intégrateur de système (SI) conçoit l'architecture de contrôle-commande en choisissant les composants, matériels et logiciels, qui respectent le P&ID mais également les contraintes formulées par le client ou utilisateur. Cependant, il doit faire face à certains défis tels que :

- La sélection des composants pertinents, permettant de considérer les exigences en termes de performances fiabilistes définies dans les spécifications du projet,
- Le choix d'une architecture parmi un large éventail d'architectures type possible pour le procédé industriel,
- La définition d'un plan de maintenance, préventif ou correctif, en adéquation avec les ressources disponibles,
- La restriction des ressources propres à la phase d'avant-vente, se caractérisant par une expertise limitée dans le domaine de la sûreté de fonctionnement et des évaluations fiabilistes couplée à des contraintes de temps pour proposer rapidement un design d'architecture et valider ses performances.

Afin de répondre aux exigences de sûreté de fonctionnement, la solution actuelle pour le SI se fonde principalement sur la mise en place de niveaux de redondance des équipements ou d'un ensemble de composants constituant l'architecture ainsi que sur la mise en place d'un plan de maintenance surdimensionné pour toute la durée d'exploitation.

Évaluer les performances de sûreté de fonctionnement de l'architecture semble être un défi non négligeable pour un SI, étant donné que le comportement du système est [1] :

- Dynamique pendant son utilisation,
- Complexe pour établir un plan de maintenance,
- Multimodal avec l'indisponibilité de certains composants pouvant avoir un impact différent sur le reste de l'architecture dans son ensemble,
- Soumis à des phénomènes de vieillissement des équipements.

Actuellement, l'approche utilisée par Schneider Electric pour évaluer les performances fiabilistes lors de la phase d'avant-vente consiste à envoyer le design de l'architecture proposé par le SI à un expert en gestion des risques et en sûreté de fonctionnement. Cet expert analyse l'architecture par le biais d'outils et de méthodologies internes, réalise des analyses comparatives pour valider ou réfuter le design proposé. Cette approche est efficace pour concevoir la meilleure architecture, mais elle nécessite des ressources considérables en moyens humains et un temps jugé trop important en phase d'avant-vente pour que l'expert puisse fournir des résultats. De plus, si l'architecture déployée ne répond pas aux spécifications du client après l'évaluation des performances faite par l'expert, il faut redéfinir le design et soumettre la nouvelle architecture à une nouvelle évaluation, ce qui implique non seulement un coût pour le SI mais aussi pour l'expert et par conséquent le client (ou l'utilisateur final).

L'autre approche repose sur une évaluation se basant sur des modèles. Les diagrammes de blocs de fiabilité (RBD) et l'analyse par des arbres de défaillance (FTA) sont parmi les modèles les plus utilisés pour évaluer les indicateurs de fiabilité. Cependant, ces modèles ne sont pas adaptés à des systèmes comme les architectures de contrôle-commande car ils ne tiennent pas compte du comportement dynamique dû à des reconfigurations possibles, à des multiples modes de fonctionnement, à la prise en compte des réparations. Ces méthodes sont également voraces en temps car pour chaque proposition d'architecture, nous devons effectuer une nouvelle analyse et reconstruire quasi intégralement le modèle pour décrire l'architecture d'un point de vue fiabiliste.

Dans le choix des modèles dynamiques, la littérature nous fournit plusieurs formalismes, allant des automates stochastiques aux réseaux de Petri [2-3], en passant par le processus de Markov booléens [4] qui combine des arbres de défaillance et des automates, et qui permettent de calculer des indicateurs fiabilistes avec des caractéristiques stochastiques. Les modèles se construisent en considérant la reconfiguration au sein du système, le multimode ou encore les phénomènes de dégradation. Néanmoins, ces formalismes nécessitent une expertise scientifique non négligeable en plus d'une contrainte de temps pour créer les modèles de l'architecture de contrôle-commande. De plus, il faut changer le modèle une fois qu'une modification est apportée au design ou aux spécifications, ce qui conduit à devoir adapter rapidement les modèles formels.

Toutes ces restrictions nous obligent à développer une nouvelle méthodologie se fondant sur la génération automatique pour aider le SI à évaluer les performances

fiabilistes d'une architecture de contrôle-commande pendant sa phase de définition sans avoir une expertise concernant les modèles formels. L'approche est basée sur les principes suivants :

- L'obtention du modèle permettant l'évaluation de la sûreté de fonctionnement doit être rapide et transparente au SI qui propose l'architecture de contrôle-commande, car celui-ci n'a potentiellement ni d'expertise dans le domaine de la sûreté de fonctionnement, ni celle des modèles formels qui sont utilisé dans ce domaine
- L'évaluation doit être réalisée dynamiquement pour tous types d'architecture,
- L'évaluation doit être réalisée rapidement avec des compétences et des ressources limitées,
- Les résultats sur les performances doivent être simple à analyser et à comprendre. Des recommandations peuvent être fournies en complément pour améliorer le niveau de fiabilité et de disponibilité.

Le défi consiste donc à trouver une solution permettant de générer rapidement et automatiquement un modèle d'architecture de contrôle-commande, quelles que soient sa taille et sa complexité. Ceci est le cœur de notre travail présenté dans cet article. Ce modèle fournira un moyen efficace d'évaluer dynamiquement les performances de sûreté de fonctionnement de l'architecture.

### *B. Explication de la méthode d'évaluation*

Un schéma global de la méthode proposée pour l'évaluation des performances de sûreté de fonctionnement est présenté dans la Fig. 2.

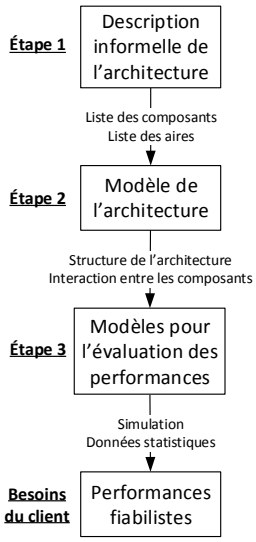


Fig. 2. Description de la méthode d'évaluation des performances fiabilistes.

Comme décrit dans les sections précédentes, l'objectif principal de nos travaux est de pouvoir calculer les indicateurs et performances fiabilistes d'une architecture de contrôle-commande dans le but de déterminer si cette dernière respecte les exigences formulées par le client (Fig. 2, Besoins du client).

Ces performances seront disponibles en exécutant le modèle de simulation dynamique qui représente le comportement fonctionnel et dysfonctionnel du système complété par une description du contexte d'utilisation de l'architecture au cours du temps. Cette combinaison est regroupée sous la forme de modèles pour l'évaluation des performances (Fig. 2, étape 3). Sur ces modèles, des données statistiques brutes concernant le comportement du système sur toute sa durée de vie seront obtenues suite à une série de simulations. Elles seront utilisées par la suite pour calculer statistiquement les performances fiabilistes de l'architecture de contrôle-commande.

Étant donné que ces modèles nécessitent une expertise et un temps de modélisation que nous ne pouvons pas avoir en avant-vente, nous devons trouver une solution pour les générer automatiquement ces modèles d'évaluation à partir de la description informelle de l'architecture de contrôle-commande. Cette génération nécessite dans un premier temps d'identifier la liste des composants du système, leurs interactions au sein du système ainsi que leurs propriétés.

Les seules informations disponibles à partir de la description informelle (Fig. 2, étape 1) donnée par les outils actuels de design pour le SI consistent en :

- La liste, le nombre et le type pour chaque composant constituant le système,
- La structure de l'architecture avec l'emplacement de tous les composants dans les différentes aires ainsi que leurs liens.

Pour faire le lien entre l'étape 1 et l'étape 3, un modèle semi-formel intermédiaire est nécessaire (Fig. 2, étape 2) dont le noyau est composé de deux éléments : la définition d'une bibliothèque pour les composants décrivant leur comportement, et la spécification de leurs interactions fonctionnelles et dysfonctionnelles.

Même si la liste de composants est large, ces composants peuvent être regroupés en familles, une famille caractérisant des types de composants fournissant le même service. Par exemple, il est possible de définir une famille pour tous les contrôleurs fournissant un service de contrôle du procédé industriel. Les composants d'une même famille peuvent avoir des performances fiabilistes différentes, même s'ils fonctionnent dans les mêmes conditions, car il est possible que l'un est techniquement plus robuste face à l'occurrence d'une panne par rapport à un autre. Cela nécessite donc la modélisation de bibliothèques configurables afin de pouvoir prendre en compte toutes ces variations mineures dans le comportement des composants ou dans leurs spécifications techniques.

Une défaillance des composants de l'architecture peut entraîner des défaillances en cascade se répercutant sur le reste des composants du système. Par exemple, si un module d'alimentation électrique d'un contrôleur s'arrête, tous les modules d'entrée/sortie contrôlés par ce dernier deviennent indisponibles. Par conséquent, la relation fonctionnelle et dysfonctionnelle entre les composants doit être incluse dans l'étape 2 et l'étape 3.

Ainsi, les exigences scientifiques qui en résultent pour construire le modèle de l'architecture et les modèles pour l'évaluation des performances sont principalement basées sur la définition de :

- Un modèle générique des différentes familles de composants qui fournissent les mêmes fonctions et/ou services avec un comportement similaire,
- Un modèle pour décrire les interactions entre les composants, dépendant également de la structure de l'architecture,
- Des algorithmes pour instancier les familles de composants à partir de la description informelle basés sur les paramètres propres à l'architecture étudiée.

## III. CONSTRUCTION DU MODELE D'EVALUATION DES PERFORMANCES FIABILISTES

Les sections suivantes décrivent la construction du modèle de l'architecture ainsi que des modèles pour l'évaluation des performances.

### *A. Construction du modèle de l'architecture*

L'UML (*Unified Modeling Language*) offre un large éventail de formalismes graphiques qui permettent de représenter différents aspects d'un système [5-6]. Ce langage de modélisation peut être utilisé comme un méta-modèle capable de décrire et de partager toutes les informations jugées utiles et liées à l'architecture. Il a été également démontré que la combinaison de l'UML avec des études de sûreté de fonctionnement, comme dans notre cas, est efficace pour obtenir des performances fiabilistes de nos systèmes [7].

C'est pourquoi, le comportement de nos architectures de contrôle-commande sera décrit avec les diagrammes UML suivants [8] :

- Diagrammes de classes (Fig. 3) pour décrire la structure de l'architecture ainsi que les liens physiques entre les composants,
- Diagramme d'états (Fig. 4) pour décrire le comportement fonctionnel et dysfonctionnel d'un composant unique, avec la liste de tous les états dans lequel il peut se trouver,
- Diagrammes de séquence (Fig. 5) pour décrire les interactions entre les composants et les conditions en amont pour déclencher les effets d'une défaillance sur le reste de l'architecture.

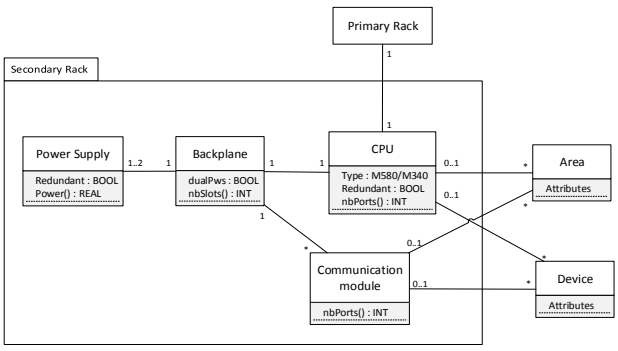


Fig. 3. Exemple de diagramme de classes pour un rack secondaire.

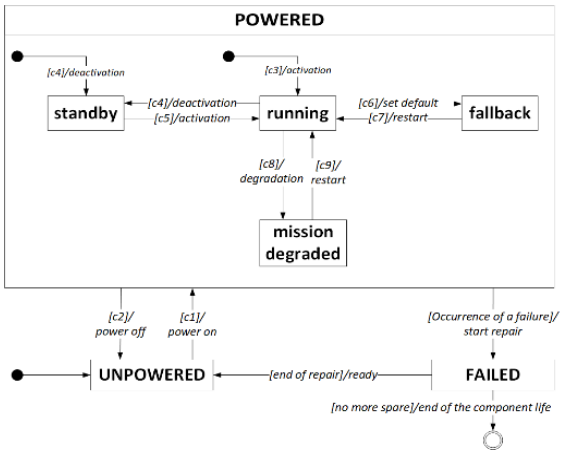


Fig. 4. Diagramme d'états général pour tous types de composants dans une architecture de contrôle-commande.

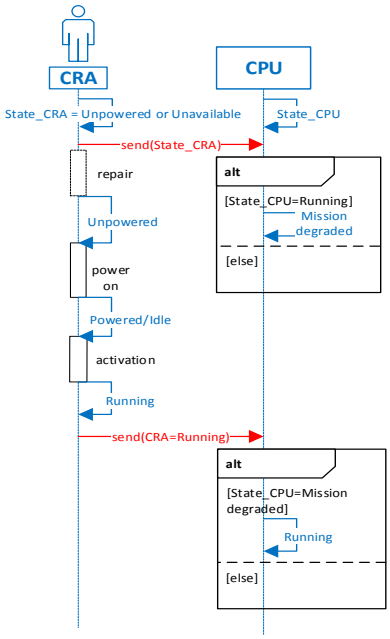


Fig. 5. Exemple de diagramme de séquence décrivant la communication entre le module Ethernet (CRA) et le processeur (CPU).

### *B. Construction des modèles pour l'évaluation des performances*

La prochaine étape consiste à définir le mécanisme formel qui permet de transformer le modèle d'architecture représenté par les diagrammes UML présentés dans la section précédente en un modèle d'évaluation des performances, dynamique et formel.

Parmi les différents types de modélisation qui admet une représentation dynamique d'une architecture de contrôle-commande dans le but d'évaluer ses performances fiabilistes, les réseaux de Petri colorés (RdPC) [9-10] associés à la simulation de Monte-Carlo ont été sélectionnés. En effet, ce formalisme couplé à la sémantique des RdP sont similaires aux diagrammes UML utilisés. Le RdPC reste également un langage de modélisation des systèmes à événements discrets combinant les capacités des réseaux de Petri avec un langage de programmation de haut niveau. Ces caractéristiques sont déterminantes pour la construction d'un modèle RdP en combinant plusieurs modèles de RdP pour chaque famille de composants. De plus, le logiciel CPNTools [11] permettant

la modélisation et la simulation des RdPC est un logiciel libre d'utilisation.

Dans le cas de la simulation de Monte-Carlo, les modèles RdPC utilisent une série de moniteurs qui permettent de récupérer des données brutes issues des simulations, telles que les événements lorsque le système est disponible ou non, et qui seront utilisées ensuite (en post-traitement) pour l'évaluation statistiques des indicateurs d'intérêt de la sûreté de fonctionnement. L'analyse statistique de ces données fournira ainsi les indicateurs de sûreté de fonctionnement que nous fournirons au SI (disponibilité, fiabilité, temps moyen avant défaillance (MTTF), etc.) [12-13].

*1) Description du cas d'étude*

Pour bien comprendre les modèles de RdP, nous allons considérer le cas d'étude donné sur la Fig. 6 qui décrit une architecture de base comprenant un large éventail de composants que l'on peut trouver sur une architecture de contrôle-commande. Il est composé d'un processeur gérant plusieurs équipements et modules d'entrée/sortie, dont certains sont situés dans la même aire que le processeur et les autres dans deux aires contrôlées à distance :

- Un processeur (CPU),
- Six modules d'entrées-sorties avec deux pour chaque rack,
- Trois modules d'alimentation électrique,
- Trois supports de rack,
- Deux modules Ethernet (CRA) pour connecter les deux aires à distance avec le processeur,
- Trois modules de communication pour connecter les équipements terminaux et le processeur à la salle de contrôle,
- Quatre équipements terminaux (variateurs de vitesse et contrôleurs de moteur).

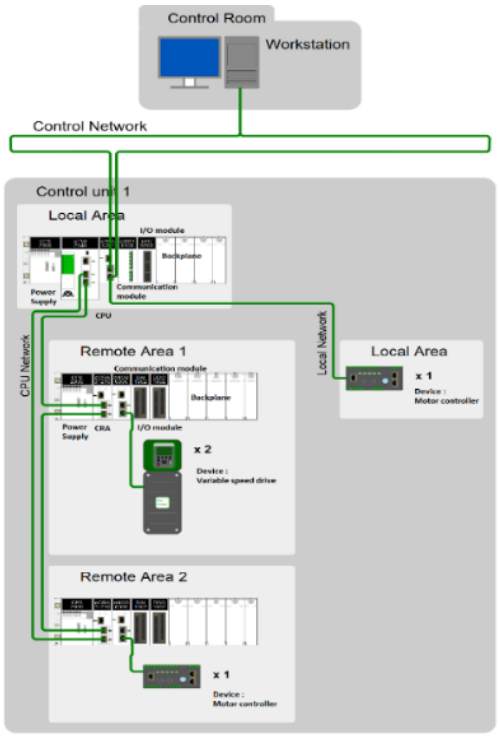


Fig. 6. Description informelle du cas d'étude.

La description informelle de ce cas d'étude donne également le nombre d'aires ainsi que la liste des composants dans chaque aire, comme suit :

- Une unité de contrôle (Control Unit) rassemblant tous les composants qui sont gérés par le même processeur,
- Une aire locale (Local area) qui comprend le rack où se situe le processeur,
- Deux aires à distance (Remote areas) pour le reste des modules et équipements terminaux.

À partir de cette étude de cas, nous montrerons comment les RdP ont été obtenus pour décrire la structure et le comportement d'une architecture de contrôle-commande complète à l'aide des diagrammes UML. Ces réseaux de Petri seront un modèle de base fixe, valable pour toutes les structures possibles d'architecture, seule l'instanciation des paramètres de l'architecture étudiée par le SI changera pour être adaptée à la configuration actuelle proposée. Cette instanciation se fait à partir de jetons colorés décryptant la configuration avec la liste des composants, les interactions des composants et la structure de l'architecture.

Dans un premier temps, nous récupérons les informations de l'architecture à partir de la description informelle, puis nous mettons en place l'instanciation du modèle pour avoir la configuration de l'architecture, ensuite, après avoir exécuté la série de simulations sur les modèles RdP, nous faisons tourner l'algorithme de post-simulation pour récupérer les performances fiabilistes de l'architecture.

*2) Instanciation de la configuration de l'architecture dans le modèle d'évaluation de performances*

La première étape pour mettre en place l'évaluation de sûreté de fonctionnement est de configurer le modèle RdP de l'architecture de contrôle-commande à partir de sa description informelle ainsi que des diagrammes UML. Comme dit précédemment, la description informelle donne les informations suivantes :

- La liste des composants constituant le système, classés par famille,
- Le nombre de composants pour chaque aire de l'architecture,
- Les canaux de communication et les connections entre les composants,
- Un tableau avec les caractéristiques en lien avec la fiabilité pour chaque composant (MTBF, MTTR…).

Ces données sont instanciées dans plusieurs jetons colorés permettant d'attribuer à chaque composant un modèle de la famille lui correspondant avec les informations concernant sa localisation et les différentes communications que ce dernier partage avec le reste des composants de l'architecture. Certaines familles de composants peuvent faire l'objet d'une récupération de jetons supplémentaires pour fournir uniquement des informations sur les aires dans le système (Control Unit, area, rack, etc.) afin de compléter le modèle décrivant le comportement du composant. La Fig. 7 illustre l'instanciation des données dans le modèle RdP.

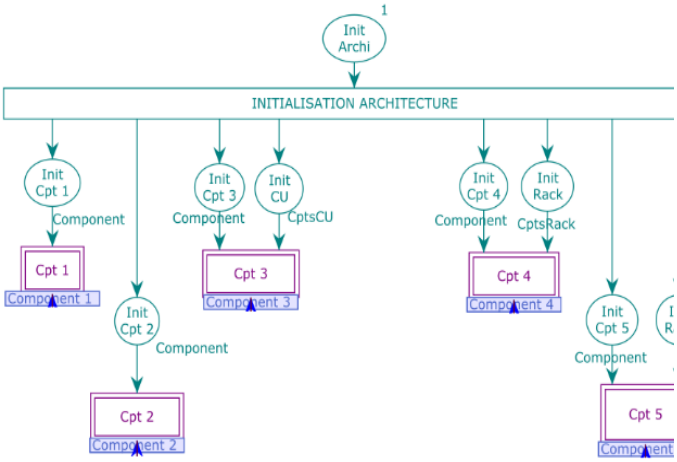


Fig. 7. Instanciation du RdP.

L'utilisation mixte de représentations paramétriques et de jetons structurés a été appliquée pour la génération automatique de RdP et l'instanciation dans le contexte de l'évaluation des performances temporelles d'une architecture de contrôle [14].

*3) Description des familles de composants dans le modèle d'évaluation de performances*

Une fois que la liste des composants est instanciée à partir de la description informelle et que chaque composant est configuré dans le modèle RdP approprié, la série de simulations de l'architecture peut commencer à être exécutée. Comme il a été dit plus haut, chaque famille de composants se comporte différemment en fonction de ses caractéristiques techniques, par exemple un processeur n'a pas les mêmes états qu'une alimentation électrique.

La Fig. 8 présente le modèle d'une famille de composants. Il intègre les différents états possibles en se basant sur le diagramme d'états du modèle UML développé. Il intègre également les conditions intrinsèques au composant pour changer son état ainsi que l'impact des autres composants que nous retrouvons dans le diagramme de séquence du modèle de l'architecture. Ceci est réalisé grâce à un jeton coloré qui contient le changement survenu dans l'architecture globale. Par ailleurs, il envoie également un jeton coloré lorsqu'un composant change d'état pour informer le reste de l'architecture. Outre le fait de décrire la place du composant dans l'architecture, le modèle calcule l'instant de la prochaine défaillance du composant en fonction de son taux de défaillance fourni grâce au MTTF. En effet, pour un composant électronique, ce qui est le cas pour les composants dans une architecture de contrôle-commande, dans la plupart de cas son taux de défaillance est égal à 1/MTTF.

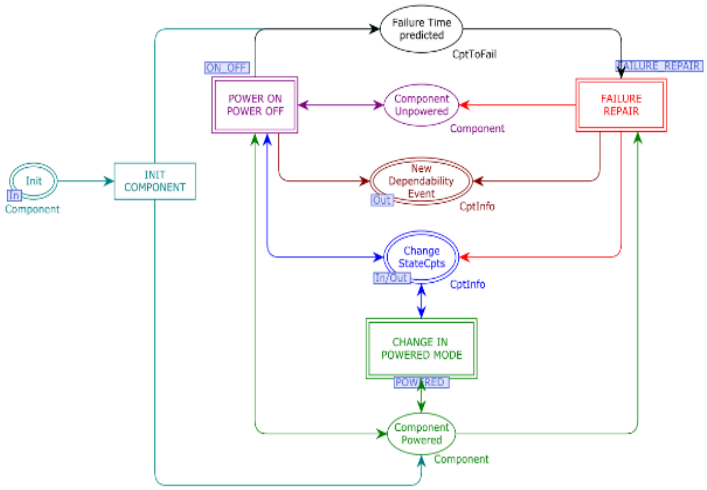


Fig. 8. RdP pour une famille de composant dans l'architecture de contrôle-commande.

Ce modèle s'appuie sur le diagramme d'états et énumère tous les états dans lesquels le composant peut être : « alimenté » (Powered), divisé en trois sous-états possibles, contenus dans le jeton étant dans la place « Component Powered » (Fig. 8), qui sont « en cours d'exécution » (Running), « mission dégradée » (Mission Degraded) et « position de repli » (Fallback) ; « non alimenté » (Unpowered) et « en panne » (Failed) que nous retrouvons dans la transition de substitution « FAILURE-REPAIR » (Fig. 8). L'état « en redondance » (Standby), qui caractérise les composants en redondance, est optionnel et dépend de la configuration du composant à l'instanciation, s'il est redondé ou non dans l'architecture.

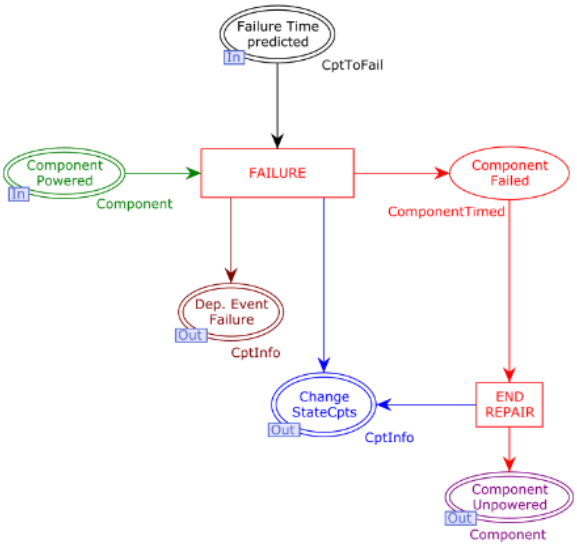


Fig. 9. RdP décrivant le comportement d'un composant après l'occurrence d'une défaillance et la fin d'une réparation.

Les conditions dans lesquelles le composant va changer d'état après un événement impactant ses performances fiabilistes sont décrites dans plusieurs sous-modèles qui sont les transitions de substitution vues dans la Fig. 8. Par exemple, dans la Fig. 9, le modèle CPN décrit un modèle propre à un composant dans le cadre de l'occurrence d'une défaillance indépendante due au composant même. Ici, seulement l'état précédent du composant en « Powered » et le

temps prévu avant sa défaillance sont pris en compte pour faire basculer le composant dans l'état « Failed ».

Le changement d'état peut être déclenché par un autre composant de l'architecture. Par exemple, la défaillance d'un module d'alimentation va mettre hors tension le processeur placé dans le même rack que le module d'alimentation. La Fig. 10 montre le modèle RdP pour mettre hors tension un composant (le composant 3 est mis hors tension suite aux défaillances des composants 3 ou 4). Ces types d'interaction, définis dans les diagrammes de séquence, sont principalement décrits par des conditions enregistrées en amont des transitions pour le changement. Après le changement, l'événement est sauvegardé par un moniteur avec l'instant d'occurrence de l'événement, la famille et l'emplacement du composant source de l'événement.

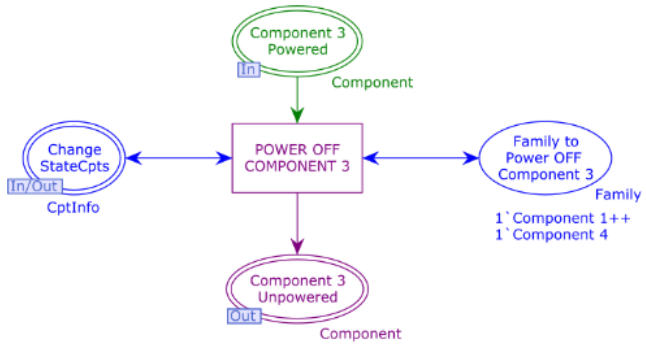


Fig. 10. RdP décrivant le comportement du composant n°3 se mettant en hors tension à cause du composant n°1 ou n°4.

Tous les événements d'intérêt pour l'évaluation de la sûreté de fonctionnement qui se produiront pendant les simulations sont récupérés dans un fichier externe : cette liste d'événements représente les données nécessaires au post-traitement pour calculer les indicateurs de sûreté de fonctionnement souhaités par le SI et le client (Fig. 11).

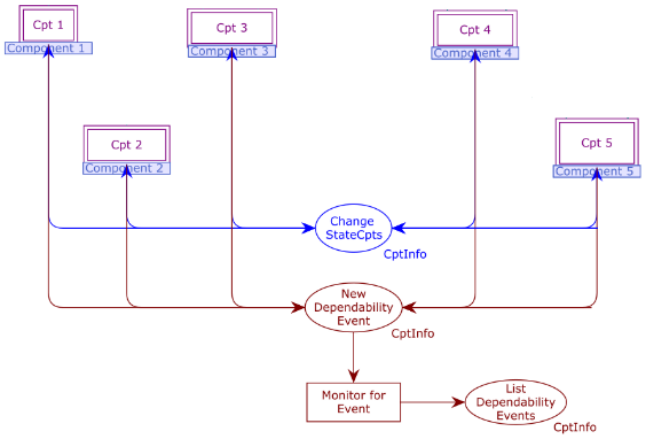


Fig. 11. RdP décrivant les interactions entre tous les familles de composants dans une architecture de contrôle-commande par le biais de la place « Change StateCpts » ainsi que l'enregistrement de tous les événements passés dans la place « List Dependability Events ».

### C. *Présentation des performances fiabilistes de l'architecture de contrôle-commande*

Après une série de simulations sur les modèles RdP (modélisant le comportement du système) présentés précédemment, un algorithme de post-traitement est effectué à partir de la liste d'événements sauvegardés dans un fichier externe. Cet algorithme calcule les indicateurs et performances souhaités par le client à partir des données brutes des événements (instants de panne, réparation, coupure de courant, etc.) survenus.

Pour prouver que les modèles RdP sont conformes à la représentation du comportement de l'architecture, nous avons comparé les performances obtenues avec les RdP par rapport à celles obtenues à partir d'un modèle statique basé sur les diagrammes de fiabilité (RBD) actuellement utilisé par Schneider Electric. Ce RBD est un modèle déjà validé avec les architectures de contrôle-commande que le SI peut proposer. Dans le tableau suivant, nous comparons les performances pour le cas d'étude décrit précédemment.

TABLE I. COMPARAISON DES INDICATEURS DE SURETE DE FONCTIONNEMENT POUR LE CAS D'ETUDE ENTRE LES MODELES RDP ET LES RBD (DUREE D'OPERATION : 10 ANS)

| *Type d'indicateurs* | *RBD* | *RdP* |
|---|---|---|
| Disponibilité (%) | 99,9598 | 99,9689 |
| MDT (en heures) | 10,93 | 10,77 |
| MTTF (en heures) | 65 126 | 66 666 |

Ces indicateurs peuvent être récupérés grâce à des modèles statiques ou dynamiques dans un délai raisonnable. Cependant, ils ne sont pas suffisants pour une analyse complète du comportement de l'architecture dans une optique d'étude de la sûreté de fonctionnement. De plus, pour un non-expert, ces chiffres peuvent être difficiles à comprendre et à analyser, de sorte que les décisions prises peuvent ne pas refléter une stratégie qui soit adéquate à la situation et adaptée pour l'utilisateur final. Par exemple, l'architecture peut ne pas être validée car elle ne respecte pas les seuils minimaux imposés et ceci pourrait impliquer un changement total du système alors qu'avec une analyse un peu plus fine, nous aurions pu constater que le changement d'un seul composant suffit pour répondre aux exigences demandées. C'est pour cette raison que nous fournissons des indicateurs supplémentaires, qui ne sont disponibles qu'avec une approche dynamique, combinée à une simulation de Monte-Carlo. La Fig. 12, la Fig. 13 et la table II sont quelques exemples de ces nouveaux indicateurs supplémentaires que nous n'avons pas pu fournir avec le RBD seul mentionné précédemment.

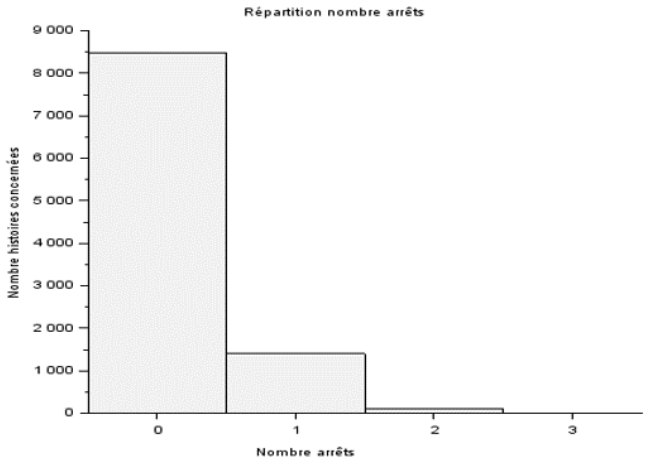


Fig. 12. Nombre d'arrêts de l'architecture complète (10 000 histoires, durée d'opération : 10 ans).

La Fig. 12 représente l'histogramme du nombre d'arrêts survenus dans une histoire de la série de simulations, ici nous voyons pour le cas d'étude que, dans la majorité des cas, il n'y a pas d'arrêt sur les 10 ans d'opération de l'architecture. Le maximum est de trois arrêts, mais n'a été retrouvé qu'une seule fois. La Fig. 13 représente l'histogramme du temps d'arrêt de l'architecture en heures dans une histoire, qui est corrélé avec la figure précédente car elle concerne le même événement à savoir l'arrêt du système. La table 2 montre l'impact de certains composants liés à l'alimentation électrique dans les aires de l'architecture : on constate ici que les supports du rack sont la principale source d'arrêt.

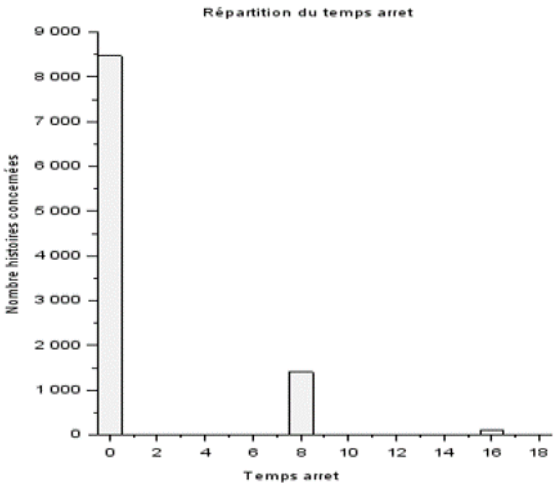


Fig. 13. Temps d'arrêt de l'architecture complète en heures (10 000 histoires, temps d'opération : 10 ans).

Dans la série d'indicateurs supplémentaires que nous donnons à travers le cas d'étude, nous voyons clairement que les décisions que le SI prendra seront plus précises, et l'aide apportée par ceux-ci sont primordiaux par rapport aux indicateurs donnés uniquement avec un modèle statique. Dans l'exemple, si les arrêts sont interdits ou doivent être réduits au maximum pour le procédé industriel, le SI peut en déduire dans le plan de maintenance, que les pièces de rechange pour les supports de rack sont une priorité ou qu'il serait judicieux de changer ces supports avec une gamme plus robuste.

L'affichage de l'évolution des indicateurs dans le temps fournit une aide non négligeable au SI pour ses décisions qu'il doit prendre en compte pour le design de l'architecture afin d'atteindre les exigences du client énoncées lors de la phase d'avant-vente.

TABLE II. IMPACT DE CERTAINES FAMILLES DE COMPOSANTS DANS L'ARRET PARTIEL OU COMPLET DE L'ARCHITECTURE DE CONTROLE-COMMANDE

| ***Famille de composants*** | ***Localisation*** | ***Part dans l'arrêt (%)*** |
|---|---|---|
| Module d'alimentation électrique | Aire locale | 18,5 |
| Module d'alimentation électrique | Aire à distance | 17,1 |
| Support de rack | Aire locale | 30,8 |
| Support de rack | Aire à distance | 33,6 |

## IV. PERPECTIVES DE L'EVALUATION DES PERFORMANCES FIABILISTES

La base du modèle pour l'évaluation des performances fiabilistes est mise en place. Les prochaines étapes consistent à compléter le modèle RdP avec de nouvelles propriétés pour les composants, à mettre à jour et à maintenir le modèle.

Nous allons par exemple améliorer le modèle RdP proposé pour avoir un modèle générique intégrant la redondance entre certains types de composants, tels que la redondance des alimentations électriques ou des processeurs, et de prendre en compte plusieurs phénomènes tels que le phénomène de vieillissement ou l'impact de l'environnement sur le comportement des composants. De même, nous envisageons d'intégrer les processeurs propres à la sécurité fonctionnelle.

Tout ceci ouvre une nouvelle voie à la génération automatique de modèles de RdP à partir d'une connaissance structurée.

## V. CONCLUSION

Cet article présente nos travaux dans le cadre de l'évaluation des performances de sûreté de fonctionnement dédié aux architectures de contrôle-commande. Nous souhaitons évaluer les indicateurs tels que la disponibilité et la fiabilité mais également des indicateurs plus spécifiques tels que l'importance de différents composants sur les indicateurs de sûreté de fonctionnement de l'architecture globale de contrôle-commande, qui aideront l'intégrateur de système (SI) à concevoir l'architecture pendant la phase d'ingénierie avant la vente. Il est basé sur des modèles de réseau de Petri coloré (RdP) pour la description du comportement du système. Les résultats de la simulation de Monte-Carlo devraient servir d'outil d'aide à la décision pour le SI. Les modèles RdP développés ont été utilisés car ils constituent actuellement le moyen le plus efficace de décrire une architecture en tenant compte du comportement dynamique et des limites en matière d'expertise et de ressources dans la phase d'avant-vente. Avec le développement du modèle RdP et ses besoins futurs, nous enrichirons dans nos futurs travaux la description de l'architecture avec d'autres caractéristiques tels que la redondance et les phénomènes de vieillissement.

## REFERENCES


[1] Castaneda G.P, J.F. Aubry, and N. Brînzei (2011). Stochastic hybrid automata model for dynamic reliability assessment. *Journal of Risk and Reliability*, 225(1), 28-41.

[2] Dutuit Y., Signoret J-P., and Thomas P. (2016). Prise en compte des transitions dynamiques au sein des réseaux de Petri stochastiques. *20e Congrès de maîtrise des risques et de sûreté de fonctionnement, LambdaMu 20, Saint-Malo, France, octobre 2016*.

[3] Aubry J.F., N. Brînzei, and M. Mazouni (2016). *Systems Dependability Assessment: Benefits of Petri Net Models*. Wiley.

[4] Bouissou M. (2009). Using BDMP (Boolean logic Driven Markov Processes) for multi-state system analysis. 6th International Conference on Mathematical Methods in Reliability, MMR 2009, Russia.

[5] Booch G., J. Rumbaugh, and I. Jacobson (1999). *The Unified Modeling Language*. User Guide.

[6] Object Management Group (2015). About the Unified Modeling Language specification version 2.5. Retrieved from https://www.omg.org/spec/UML/2.5.

[7] David P., V. Idasiak, and F. Kratz (2010). Reliability of complex physical systems using SysML. *Reliability Engineering & System Safety*, 95(4), 431-450.

[8] Boyer G., J.F Pétin, N. Brinzei, J. Camerini, and M. Ndiaye (2019). Toward generation of dependability assessment models for industrial control system. *2019 International Conference on Information and Digital Technologies (IDT), Slovakia,* 50-59.

[9] Jensen K. (1997). *Coloured Petri Nets: Basic Concepts, Analysis Methods And Practical Use Vol. 1.* Springer.

[10] CPN Tools, A tool for editing, simulating, and analyzing Colored Petri nets. Retrieved from http://cpntools.org/.

[11] Jensen K, and L. Kristensen (2009). *Coloured Petri Nets: Modelling and Validation of Concurrent Systems.* Springer.

[12] Labeau P.E., and E. Zio (2002). Procedures of Monte Carlo transport simulation for applications in system engineering. *Reliability Engineering and System Safety*, vol. 77, 217-228.

[13] Zio E. (2013). *The Monte Carlo Simulation Method For System Reliability And Risk Analysis.* Springer.

[14] Ndiaye M., J.F Pétin, J. Camerini and J.P Georges (2016). Practical use of coloured Petri nets for the design and performance assessment of distributed automation architecture. *International Workshop on Petri Nets and Software Engineering, PNSE'16, Poland,* pp 113-131.