Génération de séquences d'actions sûres par recherche d'atteignabilité

Thomas Cochard, David Gouyon, Jean-François Pétin

Génération de séquences d'actions sûres par recherche d'atteignabilité

Thomas Cochard [*,**] **David Gouyon** [*,**]
Jean-François Pétin [*,**]

[*] *Université de Lorraine, CRAN, UMR 7039, Campus Sciences, BP 70239, Vandœuvre-lès-Nancy Cedex, 54506, France*
[**] *CNRS, CRAN, UMR 7039, France*

Résumé : Cet article s'intéresse à l'ingénierie et la préparation des séquences d'actions pour les systèmes complexes critiques, avec pour objectif d'assurer que les séquences peuvent être opérées en toute sécurité sur le procédé. L'article montre, en se basant d'une part sur une modélisation formelle du système à l'aide d'automates à états communicants, et d'autre part sur des mécanismes de recherche d'atteignabilité, la faisabilité et les limites d'une approche de génération automatique de séquences respectant a priori les contraintes de sécurité.

Mots-clefs: Génération de séquences d'actions, recherche d'atteignabilité, model checking, automates à états communicants.

1. INTRODUCTION

Deux des caractéristiques principales des systèmes complexes critiques sont, d'une part, le nombre important d'équipements qui les composent (jusqu'à plusieurs dizaines de milliers) et, d'autre part, le niveau de criticité du procédé. De ce fait, le contrôle de tels procédés est soumis à une qualification, imposant un effort important en termes d'ingénierie, entre autres pour la préparation des séquences d'actions à mener. L'objectif d'une telle séquence est de mener le système depuis une situation initiale, caractérisée par l'état et la configuration opérationnelle de ces composants et par un ensemble de valeurs physiques, vers une situation objectif. Ces séquences sont définies comme une suite ordonnée d'actions, réalisées par un opérateur ou par le contrôle-commande, provoquant un changement sur l'état ou la configuration opérationnelle d'un équipement, et donc entrainant une évolution des valeurs des variables physiques. Une fois une séquence exécutée, le système doit avoir atteint la situation objectif prédéfinie.

Du fait des redondances organiques et fonctionnelles liées à la structure du système, plusieurs séquences peuvent être possibles pour atteindre une même situation objectif. La génération de séquence d'actions a pour objectif de démontrer qu'à partir d'une situation de départ donnée, une situation objectif peut être atteignable, et le cas échéant de proposer une séquence d'actions admissible, c'est-à-dire respectant les contraintes de sécurité a priori :

- la possibilité de changement d'état d'un équipement en fonction de l'état des autres équipements,
- des contraintes liées aux limites des variables physiques du procédé.

Le contrôle d'un procédé industriel, depuis les instruments (capteurs et actionneurs) jusqu'à la salle de commande, est basé sur des éléments automatisés, des chaînes de mesure d'éléments instrumentés, des éléments manuels actionnés par des agents de terrain, et des comptes rendus d'observations effectuées par des équipes de terrain sur des organes non instrumentés ou systèmes de surveillance déportés. Ces éléments en interaction se structurent en différents niveaux relatifs au procédé, au contrôle-commande et à la conduite, mais également en plusieurs niveaux de décomposition hiérarchique relatifs aux organes manipulés, aux fonctions, ... Dans cet article, aucune distinction n'est faite entre le contrôle d'actions manuelles et automatisées menées sur le procédé. Cette hypothèse est réaliste dans la mesure où le contrôle d'un procédé ne dépend pas de l'allocation des actions, mais porte en premier lieu sur des contraintes de sécurité à respecter.

Cet article propose, dans le cadre de l'ingénierie de systèmes, d'évaluer une approche de génération de séquences d'actions opérables sur le procédé en toute sécurité. La section 2 présente des approches existantes de modélisation et de génération de séquences. Sur cette base, la section 3 propose des principes d'évaluation de ces approches, appliqués sur un cas d'étude décrit en section 4 et discutés en section 5.

2. APPROCHES EXISTANTES DE MODÉLISATION ET GÉNÉRATION DE SÉQUENCES

La structuration du contrôle des procédés industriels en niveaux hiérarchiques a été formalisée pour les procédés batch dans le cadre de la norme ISA/S88. Parmi les différents travaux sur ces procédés (Arzen and Johnsson, 1996), la méthode orientée objet ASTRID, s'appuyant sur la norme ISA/S88 (Figure 1), repose sur un principe de description hiérarchique de l'installation et des modes opératoires en sous-ensembles matériels et fonctionnels : organes, ressources, fonctions et recettes.

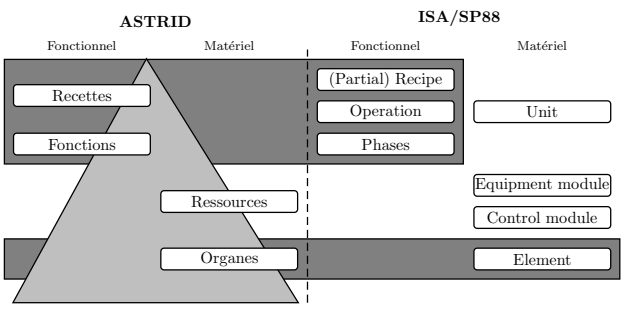

FIGURE 1. Niveaux utilisés dans la norme ISA/S88 et la méthode ASTRID

De manière complémentaire, les procédures de conduite des procédés batch ont fait l'objet de travaux de représentation graphique, comme ceux de (Arzen and Johnsson, 1996; Viswanathan et al., 1998a,b) qui introduisent le formalisme Grafchart, une extension des Grafcets, pour la modélisation séquentielle et hiérarchique de procédures. Si ces travaux permettent la représentation d'une installation et des séquences d'actions de conduite, l'obtention automatique de séquences respectant a priori des contraintes de sécurité nécessite la mise en place de mécanismes d'analyse de modèles formels. L'obtention des procédures est un problème adressé notamment par (Rivas and Rudd, 1974), qui furent parmi les premiers à proposer une approche de génération d'actions sur des vannes. Ces travaux ont par la suite été prolongés entre autres par (Foulkes et al., 1988) qui ont proposé une méthode de génération automatique de trajectoires dans des systèmes plus complexes, en se basant sur les changements d'états calculés à partir des situations initiales et attendues d'une installation, en tenant compte de règles de sécurité, sans toutefois tenir compte d'éléments structurels au niveau de l'installation ni de grandeurs physiques. D'autres approches, reposant sur l'atteinte de différents objectifs intermédiaires avant d'atteindre un objectif final, ont été proposées par (Fusillo and Powers, 1987) et (Fusillo and Powers, 1988) en vue de la planification sous contraintes de procédures (Li et al., 1997). Il s'agit principalement d'approches basées sur des heuristiques ou des techniques de recherche opérationnelle qui ne couvrent que difficilement les aspects modulaires de la modélisation (niveaux hiérarchiques, équipements, fonctions ...) et les comportements dynamiques des architectures de conduite (consignation des matériels par exemple). Les approches basées sur la théorie et les formalismes des Systèmes à Evènements Discrets apportent des réponses aux limites des approches citées ci-dessus. En effet, elles permettent de représenter de manière formelle la structure et le comportement d'une installation. Parmi ces approches, différents travaux ont déjà été menés pour la génération de séquences d'actions. Il s'agit par exemple de l'utilisation de réseaux de Petri dans les travaux de (Wang et al., 2005), qui proposent une technique de génération de procédures par synthèse, appliquée au cas d'étude utilisé par (Foulkes et al., 1988). Aucun mécanisme d'agrégation n'est cependant traité, malgré un nombre important d'éléments à considérer. Pour faciliter la représentation de l'organisation hiérarchique des systèmes considérés, les Statecharts, formalisme de description « visuelle » de systèmes complexes réactifs (Harel, 1987), peuvent alors être envisagés. Les statecharts sont une extension des diagrammes états-transitions utilisant des mécanismes supplémentaires : hiérarchie, concurrence et communication. La notion de hiérarchie étant prépondérante dans les systèmes considérés, il s'agit là d'un point fort appuyant le choix d'un formalisme de modélisation, facilitant l'agrégation des états. Cependant, le maillage inter-niveaux, c'est-à-dire le fait qu'un même équipement puisse appartenir à plus d'une fonction, ne peux être représenté en statecharts. Comme indiqué dans (Harel and Kahana, 1992), cette composition apporte trop de problèmes de sémantique formelle pour être envisagée. Ce problème semble pouvoir être résolu à l'aide des automates à états communicants (Alur and Dill, 1994), une autre extension des automates à états, qui intègrent les notions de concurrence et de synchronisation. De par leur syntaxe et sémantique, les langages formels cités ci-dessus permettent de simuler et d'analyser formellement les modèles et en particulier de vérifier des propriétés, telles que l'atteignabilité d'une situation souhaitée, par un parcours de l'espace d'états. La spécification formelle de règles, intégrées aux modèles utilisés, permet d'assurer un respect complet des règles de sécurité. Le chemin dans l'espace d'états ayant permis d'atteindre une situation objectif constitue une séquence d'actions possible, respectant ces règles. Les techniques de synthèse automatique de la commande (Ramadge and Wonham, 1987) permettent d'obtenir l'ensemble des chemins possibles (Yeh and Chang, 2012), et ont montré leur intérêt dans les travaux de (Qiu, 2005) et (Pétin, 2007) dans le cadre de la reconfiguration de systèmes manufacturiers. Elles posent cependant le problème du critère de choix du chemin à suivre parmi l'ensemble des possibles. Les approches par model-checking intègrent également la vérification de l'atteignabilité d'un état donné, et peuvent être utilisées pour générer des séquences d'actions, comme l'ont montré (Li et al., 2014) pour la génération de procédures cycliques. Par rapport aux techniques de synthèse formelle, elles ont l'avantage de proposer une seule séquence parmi les possibles, et donc de ne pas poser le problème du choix. Cet intérêt est renforcé dans la mesure où, comme l'ont montré les travaux de (Marangé et al., 2011), la séquence générée peut être très proche de l'optimum.

3. ÉVALUATION DES APPROCHES EXISTANTES

3.1 Principes mis en œuvre pour l'évaluation

Les travaux présentés dans cet article cherchent à évaluer une méthodologie pour la génération automatique de séquences d'actions depuis les schémas mécaniques de l'installation et depuis les spécifications des propriétés de sécurité à respecter, s'appuyant sur (FIGURE 2) :

- un modèle de l'installation en automates à états communicants utilisant des modèles génériques et intégrant des connaissances relatives aux conditions d'exécution des actions dans les gardes des transitions, structuré en différents niveaux,
- une formalisation de la situation objectif à atteindre sous la forme d'une propriété d'atteignabilité exprimée en logique formelle CTL,

— et un mécanisme de vérification de la propriété d'atteignabilité.

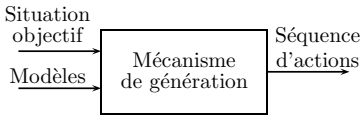

FIGURE 2. Entrées-sorties pour la génération automatique d'une séquence d'actions

3.2 Principes de modélisation utilisés

Niveaux de structuration Dans cet article, une décomposition en trois niveaux, basée sur les principes de la méthode ASTRID, est utilisée pour l'évaluation d'une méthodologie d'obtention de séquences d'actions. Le niveau « équipement » est le niveau le plus bas, correspondant à une agrégation des niveaux « organes » et « ressources » d'ASTRID, comprenant les différents équipements manipulés (vannes, pompes, ...). D'un point de vue fonctionnel, les équipements sont considérés par « fonctions » permettant l'évolution d'une ou plusieurs variables physiques appartenant à l'ensemble φ des variables physiques du procédé global. Enfin, le niveau « recette », propre au procédé, décrit l'ensemble des objectifs à atteindre.

Modélisation à l'aide d'automates à états communicants Le formalisme des automates à états communicants a été défini à l'origine par (Alur and Dill, 1994). Il a été introduit comme sous-classe des automates à états temporisés. Les automates à états communicants peuvent être définis par un 7-uplet $A = (S, X, L, T, S_m, s_0, v_0)$ tel que :

- S est un ensemble fini de localités ;
- X est un ensemble fini de variables entières ;
- L est un ensemble d'évènements décomposé en deux ensembles disjoints L_e et L_r, où
  - L_e est l'ensemble des évènements émis ;
  - L_r est l'ensemble des évènements reçus.
- T est un ensemble de transitions $(s, l, g, m, s') \in S \times L \times G \times M \times S$, où
  - G est l'ensemble des gardes (conditions sur les variables de X) ;
  - M est l'ensemble des mises à jour sur les valuations des variables.
- $S_m \subseteq S$ est un ensemble de localités marquées ;
- $s_0 \in S$ est la localité initiale ;
- $v_0 : X \leftarrow \mathbb{N}$ est la valuation initiale des variables.

Dans les modèles automates de cet article, les conventions graphiques utilisées sont les suivantes : les noms des localités sont exprimés en gras, la localité initiale est indiquée par une transition sans origine, les gardes sur les transitions sont entre crochets, les évènements sont indiqués en italique et suivis d'un « ! » ou « ? », pour représenter respectivement l'émission ou la réception, et les mises à jour sur les variables sont soulignées.

Proposition de modèles génériques d'équipements Les équipements, principalement des vannes et des pompes, sont les éléments sur lesquels un opérateur ou le contrôle-commande peuvent agir pour en changer l'état. Un équipement est caractérisé par un couple (*state*, *status*), défini de la façon suivante :

- *state* caractérise par un ensemble discret de valeurs possibles son état, (e.g. pour une vanne ses états ouverte / fermée),
- *status* s'apparente à sa configuration opérationnelle, par exemple une condamnation (pose d'un cadenas).

Deux localités sont utilisées exprimant le *state* de l'équipement, et une variable booléenne sera utilisée pour représenter son *status*. La valeur de cette variable booléenne sera définie à l'initialisation du modèle et ne pourra pas évoluer, l'objectif étant de déterminer l'atteignabilité d'une situation aux travers de différentes actions effectuées tout en tenant compte du *status* courant des équipements.

Le comportement d'une vanne est ainsi modélisé suivant le patron défini à la FIGURE 3. Pour évoluer d'une localité à une autre, une vanne doit recevoir un ordre d'ouverture ou de fermeture, modélisé par la réception d'un évènement *OuvrirVanne* ou *FermerVanne*. Les conditions de sécurité autorisant l'ouverture ou la fermeture d'une vanne sont modélisées dans les gardes par *Conditions_ouverture* et *Conditions_fermeture* sur les transitions. De plus, une vanne ne peut être commandée ou manipulée que si elle n'est pas condamnée, ce qui est modélisé par une condition sur la variable booléenne *Condamnée* également incluse dans les gardes.

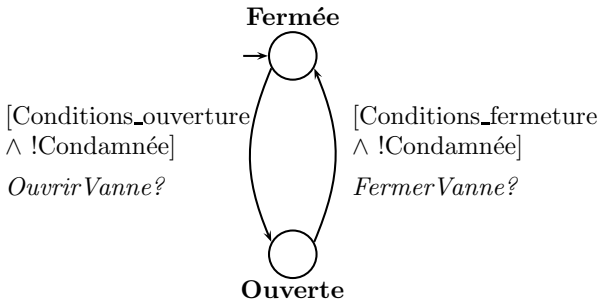

FIGURE 3. Modèle générique du comportement d'une vanne

Le comportement d'une pompe est modélisé suivant le patron défini à la FIGURE 4. Pour évoluer d'une localité à une autre, une pompe doit recevoir un ordre d'enclenchement / déclenchement, modélisé par la réception d'un évènement *EnclencherPompe* / *DéclencherPompe*. De plus, il peut exister des conditions de sécurité pour l'enclenchement / déclenchement d'une pompe, modélisé par les gardes *Conditions_enclenchement* / *Condition_déclenchement* sur les transitions. Une pompe ne peut être manipulée si elle est condamnée, ceci étant modélisé par une variable booléenne *Condamnée*.

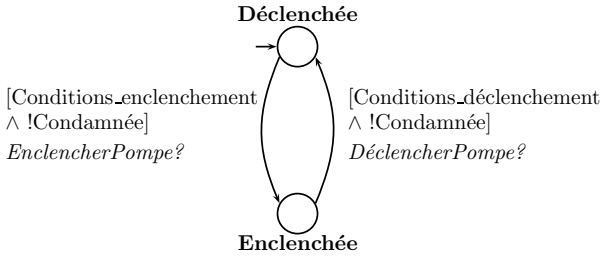

FIGURE 4. Modèle générique du comportement d'une pompe

Proposition d'un modèle générique des fonctions La configuration d'une fonction (Figure 5) est caractérisée de la manière suivante par deux localités :

- *Configurée* si $\forall eqt \in EQT_f$ le *state* de l'équipement correspond au *state* attendu pour une configuration donnée de la fonction f de l'ensemble F, provoquant une évolution sur l'ensemble $\varphi_f \subseteq \varphi$ des variables physiques concernées ;
- *Non configurée* sinon.

Le passage d'une localité à une autre est soumis d'une part à une garde correspondant aux conditions de passage liées à une configuration possible pour réaliser la fonction (notamment les states des équipements), et d'autre part à la réception d'un évènement Fonction permettant une synchronisation des différents niveaux de modèles.

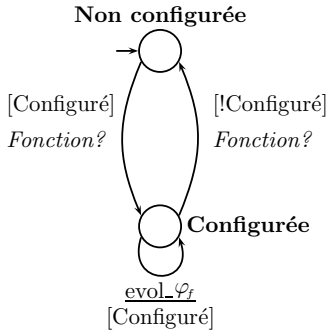

FIGURE 5. Modèle générique d'une fonction

Les différentes configurations possibles peuvent être trouvées automatiquement par un parcours de graphe représentant les équipements par des sommets et les tuyaux par des arcs.

Modélisation des recettes Le niveau « recette », décrivant l'ensemble des situations de fonctionnement possibles, est spécifique à chaque procédé. La transition d'une localité à une autre est soumise d'une part à une garde correspondant à des seuils sur les valeurs physiques et des *status* d'équipements, et d'autre part à la réception d'un évènement Recette permettant une synchronisation des différents niveaux de modèles.

Synchronisation des différents modèles L'évolution du comportement du modèle de l'installation s'effectue de la manière suivante : à réception d'un évènement l_{eqt_i} de l'ensemble $L_{eqt} \subseteq L_e$ des évènements associés aux changements de *state* d'un équipement, le modèle de ce dernier franchit une transition et change de localité active. Ces actions ont un impact sur le comportement des fonctions, et donc sur le procédé. Le principe de l'approche utilisée est donc de s'appuyer sur un automate « générateur « (FIGURE 6) ayant un double rôle :

- générer des évènements modélisant des actions effectuées au niveau des équipements : cela permet de parcourir l'ensemble de l'espace des états possibles de l'installation, et ainsi permettre la vérification de l'atteignabilité d'une situation objectif,
- synchroniser et ordonnancer l'exécution cyclique des différents modèles : émission d'un évènement d'action sur un équipement (l_{eqt_i}!), puis d'un évènement de mise à jour des fonctions (*Fonction !*), puis de mise à jour de la recette (*Recette !*).

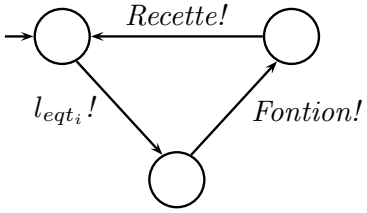

FIGURE 6. Modèle générique d'automate générateur d'évènements

Recherche d'atteignabilité d'un situation La situation objectif est caractérisée par une propriété p portant sur l'atteinte d'une localité du modèle de la recette. L'existence d'un chemin menant d'un état initial à un état dans lequel la propriété p est vérifiée s'exprime formellement en CTL par « EF p » (FIGURE 7). Ce chemin correspond à un ensemble ordonné d'événements émis (d'actions à effectuer) pour atteindre cette situation.

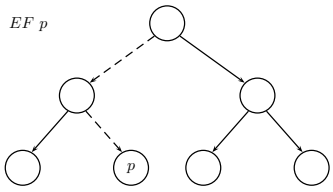

FIGURE 7. Existence d'un chemin (en pointillés) vérifiant à un moment donné une propriété p

Par une exploration de l'espace d'états, et par la vérification d'une propriété d'atteignabilité, les mécanismes de model-checking permettent de déterminer un chemin menant le procédé à la situation objectif. Il est possible de l'obtenir automatiquement à l'aide d'outils logiciels de model checking.

4. CAS D'ÉTUDE CISPI

4.1 Présentation du cas d'étude

Á des fins d'évaluation, le cas d'étude considéré est basé sur la plate-forme CISPI (FIGURE 8) du Centre de Recherche en Automatique de Nancy,

dédiée à la Conduite Interactive et Sûre de Procédés Industriels (http://safetech.cran.univ-lorraine.fr/). Avec les différentes redondances physiques du procédé, il existe sur cette plateforme 8 configurations différentes pour approvisionner la bâche 002BA à partir de la bâche 001BA.

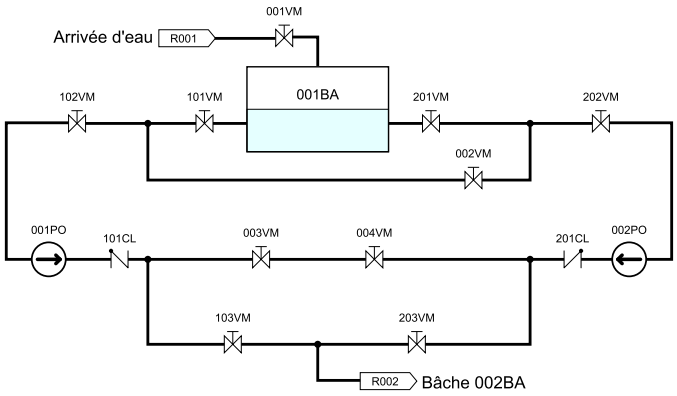

FIGURE 8. Plan de circulation des fluides de la plate-forme CISPI

Les conditions de sécurité à respecter sont relatives :

- d'une part à la mise en route des pompes, qui nécessite au préalable l'ouverture de vannes amont et avales,
- et d'autre part à la fermeture de vannes, qui peut nécessiter l'arrêt préalable de pompes.

Les modèles génériques présentés précédemment ont été instanciés pour modéliser les 10 vannes et 2 pompes considérées, ainsi que pour les 8 configurations (fonctions) possibles. Ces modèles ont été implémentés à l'aide du logiciel Uppaal (FIGURE 9). Cet outil intègre une interface graphique d'édition d'automates à états communicants et un model checker permettant notamment la vérification de propriétés d'atteignabilité. Parmi l'ensemble des model checkers disponibles (NuSMV, SPIN, ...), celui-ci a été retenu pour sa facilité de prise en main, l'objectif étant d'évaluer la faisabilité de ce type d'approche.

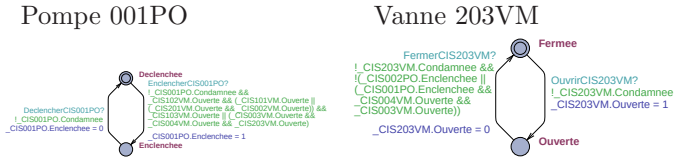

FIGURE 9. Exemples de modèles instanciés extraits de la plate-forme CISPI

L'extrait de la recette de la FIGURE 10 montre deux localités *SF0* et *SF1* correspondant à des situations de fonctionnement. La transition de l'une à l'autre est conditionnée par l'atteinte d'un niveau dans la bâche 002BA. La propriété CTL permettant alors de déterminer une séquence d'actions pour atteindre la localité *SF1* s'écrit *EF recette.SF1*.

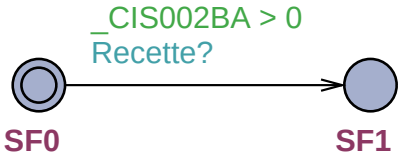

FIGURE 10. Extrait du modèle de la recette

4.2 Résultats de la génération de séquences

Nous considérons une première hypothèse selon laquelle la plate-forme est « à l'arrêt », c'est-à-dire que toutes les vannes sont fermées et que les pompes sont déclenchées. De plus, on considérera que le niveau de la cuve 001BA ne constitue pas une limitation pour l'alimentation de la bâche 002BA. La TABLE 1 donne une séquence possible, permettant l'alimentation via la » voie de gauche « . Il est à noter que l'ordre dans lequel ces actions sont données est primordial car il permet d'assurer le respect a priori des contraintes de sécurité intégrées au modèle.

TABLE 1. Séquence d'actions 1 : pas de condamnation

	Action	Équipement
1	Ouvrir	101VM
2	Ouvrir	102VM
3	Ouvrir	103VM
4	Enclencher	001PO

Dans l'hypothèse où la vanne 201VM est *condamnée* à l'état *fermée* et qu'une maintenance de la pompe 001PO est envisagée (impliquant alors la condamnation de la vanne d'isolement 102VM à l'état *fermée*), la TABLE 2 donne une séquence alternative, permettant également d'atteindre l'objectif défini.

TABLE 2. Séquence d'actions 2 : maintenance de 001PO, condamnation de 102VM et 201VM

	Action	Équipement
1	Ouvrir	203VM
2	Ouvrir	202VM
3	Ouvrir	101VM
4	Ouvrir	002VM
5	Enclencher	002PO

5. DISCUSSION

La mise en œuvre sur un cas d'étude de laboratoire a mis en évidence la possibilité de générer des séquences d'actions par recherche d'atteignabilité, avec une approche basée sur des méthodes existantes. Cependant, l'augmentation de la taille des modèles en vue de la mise à l'échelle d'une approche de ce type montre rapidement ses limites (FIGURE 11). En effet, l'application sur des exemples de taille supérieure montre que l'espace d'état parcouru croit de manière très rapide. De plus, ces séquences ne sont pas optimales, allant même jusqu'à faire apparaître des actions « inutiles », comme par exemple une succession d'ouverture/fermeture sur un même équipement.

Le parcours en largeur de l'espace d'état permet de limiter la taille de la séquence générée, et même de générer des séquences proche de l'optimalité. Cette technique est cependant très sensible au nombre d'éléments modélisés. Ainsi, la taille de l'espace d'état augmente de façon exponentielle avec le nombre d'équipements.

Le parcours en profondeur permet de limiter le phénomène d'explosion combinatoire, avec toutefois la contrepartie de générer des séquences composées d'un nombre important d'actions. Bien que le nombre d'état calculés semble augmenter de façon linéaire, cette méthode ne serait donc pas adaptée pour une application réelle.

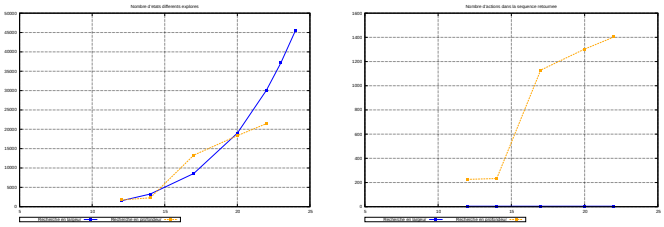

FIGURE 11. Évolution de la taille de l'espace d'états parcourus / du nombre d'actions composant la séquence générée en fonction du nombre d'équipements pris en compte

6. CONCLUSION ET PERSPECTIVES

Cet article propose une évaluation d'une approche de génération automatique de séquences d'actions sûres a priori, utilisant un mécanisme de recherche de l'atteignabilité d'une situation souhaitée dans un réseau structuré d'automates communicants. Les résultats obtenus permettent finalement de mettre en évidence les problèmes de cette approche quant à son applicabilité sur des installations de taille plus importantes.

Différents axes sont par conséquent actuellement à l'étude en vue de réduire l'explosion combinatoire engendrée par le parcours de l'espace d'état, notamment d'un point de vue de la modélisation. D'une part, des techniques d'agrégation ou de modélisation multi-échelle devraient permettre de réduire la taille des modèles, et donc de l'espace d'état possible. D'autre part, une intégration plus forte de contraintes métier devrait également permettre de limiter la taille de l'espace d'états parcouru pour trouver une séquence admissible.

7. RÉFÉRENCES

Alur, R. and Dill, D.L. (1994). A theory of timed automata. *Theoretical computer science*, 126(2), 183–235.

Arzen, K.E. and Johnsson, C. (1996). Object-oriented sfc and isa-s88. 01 recipes presented at the world batch forum. *ISA transactions*, 35(3), 237–244.

Foulkes, N., Walton, M., Andow, P., and Galluzzo, M. (1988). Computer-aided synthesis of complex pump and valve operations. *Computers & Chemical Engineering*, 12(9), 1035–1044.

Fusillo, R. and Powers, G. (1987). A synthesis method for chemical plant operating procedures. *Computers & chemical engineering*, 11(4), 369–382.

Fusillo, R. and Powers, G. (1988). Operating procedure synthesis using local models and distributed goals. *Computers & Chemical Engineering*, 12(9), 1023–1034.

Harel, D. (1987). Statecharts : A visual formalism for complex systems. *Science of computer programming*, 8(3), 231–274.

Harel, D. and Kahana, C.A. (1992). On statecharts with overlapping. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 1(4), 399–421.

Li, H.S., Lu, M.L., and Naka, Y. (1997). A two-tier methodology for synthesis of operating procedures. *Computers & chemical engineering*, 21, S899–S903.

Li, J.H., Chang, C.T., and Jiang, D. (2014). Systematic generation of cyclic operating procedures based on timed automata. *Chemical Engineering Research and Design*, 92(1), 139–155.

Marangé, P., Pétin, J.F., Manceaux, A., and Gouyon, D. (2011). Contribution à la reconfiguration des systèmes de production : ordonnancement par recherche d'atteignabilité. *Journal Européen des Systèmes Automatisés*, 45(1/3), 45–60.

Pétin, J.F. (2007). *Méthodes et modèles pour un processus sûr d'automatisation*. Ph.D. thesis, Université Henri Poincaré-Nancy I.

Qiu, R.G. (2005). Virtual production line based wip control for semiconductor manufacturing systems. *International Journal of Production Economics*, 95(2), 165–178.

Ramadge, P.J. and Wonham, W.M. (1987). Supervisory control of a class of discrete event processes. *SIAM journal on control and optimization*, 25(1), 206–230.

Rivas, J.R. and Rudd, D.F. (1974). Synthesis of failure-safe operations. *AIChE Journal*, 20(2), 320–325.

Viswanathan, S., Johnsson, C., Srinivasan, R., Venkatasubramanian, V., and Ärzen, K.E. (1998a). Automating operating procedure synthesis for batch processes : Part i. knowledge representation and planning framework. *Computers & chemical engineering*, 22(11), 1673–1685.

Viswanathan, S., Johnsson, C., Srinivasan, R., Venkatasubramanian, V., and Ärzen, K.E. (1998b). Automating operating procedure synthesis for batch processes : Part ii. implementation and application. *Computers & chemical engineering*, 22(11), 1687–1698.

Wang, Y.F., Chou, H.H., and Chang, C.T. (2005). Generation of batch operating procedures for multiple material-transfer tasks with petri nets. *Computers & chemical engineering*, 29(8), 1822–1836.

Yeh, M.L. and Chang, C.T. (2012). An automata-based approach to synthesize untimed operating procedures in batch chemical processes. *Korean Journal of Chemical Engineering*, 29(5), 583–594.

8. BIOGRAPHIE

Thomas Cochard est actuellement doctorant au Centre de Recherche en Automatique de Nancy (UMR CNRS UL 7039), où il effectue sa thèse sous la co-direction de Jean-François Pétin et David Gouyon dans le domaine de la sûreté de fonctionnement système. Ses travaux de thèse portent sur l'utilisation d'outils formels pour l'aide à la génération de séquences sûres de fonctionnement. Il est titulaire d'un Master en Ingénierie de Systèmes Complexes. Il est également chargé d'enseignements à l'Université de Lorraine.

David Gouyon. Maître de Conférences à l'Université de Lorraine, et membre du Centre de Recherche en Automatique de Nancy (UMR CNRS UL 7039). Il effectue sa recherche au sein du département Ingénierie des Systèmes Eco-Techniques, sur des thématiques relatives à l'Ingénierie des Systèmes Automatisés. Il est impliqué dans un Master en Ingénierie de Systèmes Complexes pour lequel il est référent d'un parcours en Ingénierie Numérique des Systèmes de Production, et membre de l'AFIS.

Jean-François Pétin a obtenu le grade de docteur et l'Habilitation à Diriger des Recherches de l'Université de Nancy respectivement en 1995 et 2007. Il est actuellement professeur des Universités à l'Université de Lorraine et au Centre de Recherche en Automatique de Nancy (CRAN). Son domaine de recherche est la modélisation et la vérification des Systèmes à Événements Discrets avec un accent particulier sur la sûreté de fonctionnement des systèmes de commande critiques. Il a été impliqué dans des projets de recherche auprès de grandes sociétés dans les secteurs de l'énergie et du ferroviaire. Il est l'auteur d'environ une cinquantaine de publications dans des revues et conférences internationales et a dirigé ou dirige actuellement 9 thèses de doctorat. Il est membre du comité technique TC 3.1 de l'IFAC « Computers for control ».