DISPONIBILITE DE PRODUCTION : LES OUTILS NOUVEAUX SONT ARRIVES !

PRODUCTION AVAILABILITY : NEW TOOLS ARE COMING !

Jean-Pierre SIGNORET
TDO/EXP/ARF
Centre Scientifique et Technique Jean Feger
TOTAL
64018 - PAU Cedex

Marie BOITEAU
FRACTAL System
30 ter rue de la Libertée
94300 Vincennes

Antoine RAUZY
IML/CNRS
163, avenue de Luminy
13288 Marseille Cedex 09
FRANCE

Philippe THOMAS
GFI Consulting
Parc Cadéra – Bât. 30
Avenue Ariane
33700 Mérignac

Résumé

Depuis les origines la *Sûreté de Fonctionnement* s'est principalement intéressée aux aspects *sécurité* des installations industrielles et peu à peu l'*offre* en méthodes et outils a progressé afin de satisfaire à la *demande* correspondante. Contrepartie naturelle de la sécurité, les aspects économiques sont malheureusement restés longtemps en filigrane et n'ont pas bénéficié de cette impulsion.
Cependant depuis quelques années le besoin en études probabilistes à caractère économique se renforce. Face à une demande d'études de plus en plus détaillées portant sur des paramètres probabilistes de plus en plus précis et concernant des systèmes industriels de plus en plus complexes, l'analyste se trouve quelque peu démuni lorsqu'il essaie de mettre en oeuvre les méthodes et outils développés pour traiter des seuls aspects sécurité.
Dans le domaine pétrolier par exemple, on assiste depuis dix ans à une augmentation importante de la demande d'études de productivité prévisionnelle (*disponibilité de production*) des installations en cours de conception. Les chefs de projet les utilisent comme outil d'*aide à la décision* afin de sélectionner les meilleures architectures satisfaisant à la fois aux exigences de sécurité et de rentabilité et au meilleur coût d'investissement.
TOTAL a commencé à développer des méthodes et outils visant les études de disponibilité de production dès les années 80. Le but de cet exposé est de présenter les avancées significatives obtenues ces 2 dernières années

Summary

Since the beginning the *Reliability* studies have been mainly focused on the *Safety* of the industrial installations and, step by step, the *supply* in methods and tools has increased in order to answer the corresponding *demand*. Natural counterpart of safety, economical aspects have, unfortunately, been shadowed and have not taken benefit of this movement.
However for some year past the need in economical probabilistic studies is increasing. The analyst find himself a little bit poor when he tries to use methods and tools developed for safety purposes to face a demand involving more and more detailed studies, more and more accurate probabilistic results and more complex industrial systems.
For example In the Oil &Gas field, the demand in estimating the future productivity (*production availability)* of installation in the design stage has continuously increasing for the 10 past years. Projects managers use such studies as *decision aid* in order to select the best architectures satisfying both safety and profitability requirements.
TOTAL begins to develop methods and tools devoted to production availability calculations since the early eighties. This paper aims to present the advances in progress since the 2 past years.

Prémices

Le but du domaine embrassé par la sûreté de fonctionnement est la maîtrise des risques technologiques des systèmes industriels.

Dans le domaine de la sûreté de fonctionnement parler de *risque*, revient à considérer une grandeur à deux dimensions :

- Probabilité
- Conséquences

La première de ces dimensions mesure les *chances* de voir survenir l'événement à l'origine du risque et la seconde mesure les *effets* qui en résultent.

Une telle définition est très générale car aucune hypothèse n'étant faite a priori sur la nature des conséquences elle convient aussi bien pour décrire des risques inhérents à la sécurité que des risques relatifs à l'économie des systèmes concernés.

Les paramètres principaux utilisés par les ingénieurs fiabilistes sont souvent regroupés sous le terme FMDS (**F**iabilité, **M**aintenabilité, **D**isponibilité et **S**écurité) car depuis les origines la *Sûreté de Fonctionnement* s'est principalement focalisée sur ceux-ci. Peu à peu l'*offre* en méthodes et outils a progressé afin de satisfaire à la *demande* correspondante et les aspects "sécurité" ont été - et sont encore - privilégiés.

Les aspects économiques, contrepartie naturelle de la sécurité, sont malheureusement restés longtemps en filigrane et n'ont pas bénéficié de cette impulsion. Depuis quelques années le besoin en études probabilistes à caractère économique (Disponibilité de Production) se renforce. Face à une demande d'études de plus en plus détaillées portant sur des paramètres probabilistes de plus en plus précis et concernant des systèmes industriels de plus en plus complexes, l'analyste se trouve quelque peu démuni lorsqu'il essaie de mettre en oeuvre les méthodes et outils développés pour traiter des seuls aspects sécurité.

Sécurité versus Disponibilité (de Production)

Pour mettre en lumière les raisons pour lesquelles des méthodes et outils performantes pour traiter les aspects sécurité se retrouvent mal adapté quant aux aspect économiques il suffit de comparer les caractéristiques de ces différents types d'étude.

Tout d'abord, l'utilisation qui est faite de ces études est différente : celles relatives à la sécurité servent à instruire des dossiers qui seront expertisés par des *Autorités de Sûreté* pour obtenir des autorisation alors que celles relatives la disponibilité de production sont utilisée, en interne, par les chefs de projet comme outils d'aide à la décision. Dans le premier cas, il est nécessaire de prendre en compte des hypothèse conservatives pour être assuré que le niveau de risque est acceptable alors que dans le second il

est nécessaire de rester au plus près de la réalité (estimations "*best estimate*") afin de ne pas décider d'investissements inutiles.

D'autre part, la sécurité implique généralement des événements rares avec des conséquences catastrophiques alors que la disponibilité de production implique des événements fréquents avec des conséquences minimes (car c'est le cumul de conséquences minimes qui engendre le risque économique). Comme les approximations ne fonctionnent correctement qu'avec des probabilités faibles, elles sont très efficaces pour les études de sécurité mais deviennent beaucoup trop conservatives pour pouvoir être utilisées pour les études de disponibilité de production.

En résumé, les études de disponibilité de production, nécessitent des modélisations plus proche de la réalité et des calculs moins approchés que les études de sécurité. C'est pourquoi les méthodes et outils développés pour la sécurité doivent être adaptés voire être remplacés par de nouveaux.

Corpus des Méthodes et d'Outils

Depuis maintenant plus de 50 ans tout un corpus de méthodes et d'outils ont été développés peu à peu pour réaliser les études probabilistes. Il peuvent grossièrement être regroupés en trois catégories :

- Les **modèles basiques** (Analyse fonctionnelle, AMDEC, APR, HAZOP, ...) qui permettent à la fois de comprendre comment le système étudié fonctionne, d'identifier les risques qui lui sont propres et de réaliser une première analyse élémentaire de sa SdF ;
- Les **modèles d'analyse statique** (Bloc diagramme de Fiabilité et Arbres de défaillances) qui permettent d'analyser le système étudié d'un point de vue architectural et d'identifier ses scénarios de défaillances ;
- Les **modèles dynamiques** (Graphes de Markov, Réseaux de Petri Stochastiques, Langages formels, ...) qui permettent de réaliser des études en profondeur en prenant en compte de manière rigoureuse le comportement dynamique du système étudié (réparations, reconfigurations, ...).

La troisième catégorie se rapporte à ce qui est appelé "*Processus Stochastique*" dans les ouvrages traitant de la théorie des calculs de probabilité. C'est bien entendu vers cette catégorie que nous sommes tournés pour prendre en compte les diverse contraintes inhérentes à la modélisation de la disponibilité de production des systèmes industriels.

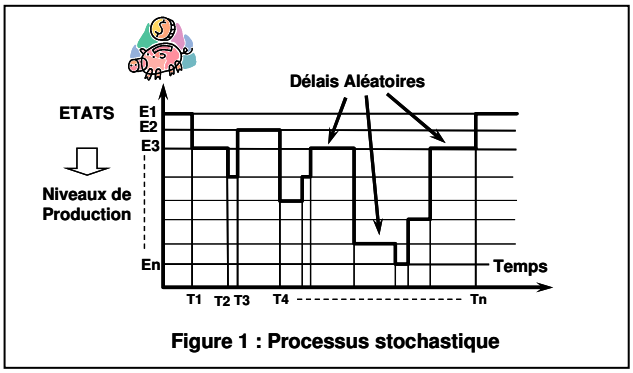


Figure 1 : Processus stochastique

La figure ci-dessus reproduit le comportement d'un système possédant plusieurs niveaux de production tel qu'on peut l'observer dans la réalité. Le temps cumulé passé dans chacun de ces niveaux permet d'évaluer la production du système sur une période donnée. Une simple division par la durée de la période considérée conduit à la disponibilité de production recherchée.

Pour évaluer correctement la disponibilité de production des systèmes industriels il nous faut donc mettre en oeuvre des modèles capable d'appréhender de tels processus stochastiques.

Graphes de Markov

Le doyen des modèles utilisé pour représenter le comportement dynamique des systèmes industriels est le "*processus e Markov*". Il est très populaire au niveau académique car il recouvre une méthode de résolution analytique (système d'équation différentielles homogène à coefficients constants) et il est facile de mise en oeuvre grâce à sa représentation graphique : le *graphe de Markov*.

Traditionnellement cette approche est utilisée pour évaluer la *fiabilité*, la *disponibilité instantanée prévisionnelle* et la *disponibilité asymptotique* des systèmes *réparables*.

Au début des années 80, un projet de recherche conjoint entre ELF et TOTAL a conduit à l'adaptation de cette approche au problèmes de disponibilité de production des plates-formes de production sous-marines.

De tels systèmes étant à la fois *multi états* (niveaux de production dépendant du nombre de puits en fonctionnement) et *multi phases* (par exemple non réparables en hiver mais réparables en été) seule une modélisation par processus stochastique était envisageable.

Grâce à leur absence de mémoire (leur futur ne dépend que de l'état présent), les processus de Markov se prêtent naturellement à la modélisation des systèmes multi phases, restait donc à traiter l'aspect multi états.

Ceci fut fait grâce au calcul des *temps moyens de séjour cumulés* passés dans les différents états (une simple intégration du vecteur de probabilité) et à l'affectation aux états d'une *efficacité* correspondant au niveau de production correspondant.

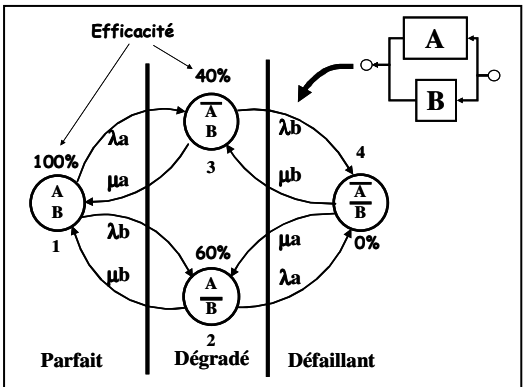


Figure 2 : Processus de Markov Multi Etats

A titre d'exemple simpl(iste)e on voit sur la figure n°2 un système formés de deux composants ***A*** et **B** ayant des capacités (ou efficacités) de production différentes. Celle de ***A*** est de 60% alors que celle de **B** n'est que de 40%. Il en résulte que les 4 états de ce système ne peuvent pas être rangée dans les deux classes dichotomiques traditionnelles "*marche / Panne*" mais doivent être séparés en 3 classes :

- "*Parfait"* (efficacité de 100%)
- *"Dégradé"* (efficacités de 60% et 40%)
- *"Panne totale*" (efficacité de 0%).

Les temps moyens de séjour cumulés pondérés par les efficacités correspondantes conduit immédiatement à la production de ce système dont la disponibilité de production se déduit de manière triviale.

Les travaux ELF/TOTAL du début des années 80 nous ont permis de développer les logiciels **MARK EXD** (mono phase) et **MARK SMP** (multi phase) qui ont par la suite été améliorés à diverses reprises. En fonction de la taille des graphes bâtissable manuellement nous les avions limité à une centaine d'états. Ces deux logiciels sont encapsulés dans notre interface de saisie graphique interactive **GRIF** qui en facilite la mise en oeuvre.

Avec l'avènement de l'atelier **ACACIA** capable de générer automatiquement de gros graphe de Markov à partir de descriptions en langage **AltaRica Data Flow** (cf ci-dessous), nous avons re-développé en 2003 le logiciel MARK EXD de manière à ce qu'il n'y ait plus de limitation en nombre d'états. Cela à donné le logiciel **MARK XPR** capable de traiter quelques millions d'états et nous y reviendrons plustard dans cette communication.

Les difficultés

Après le préambule ci-dessus, il est temps de regarder de plus près les problèmes rencontrés lorsqu'on désire modéliser en détail un système de production.

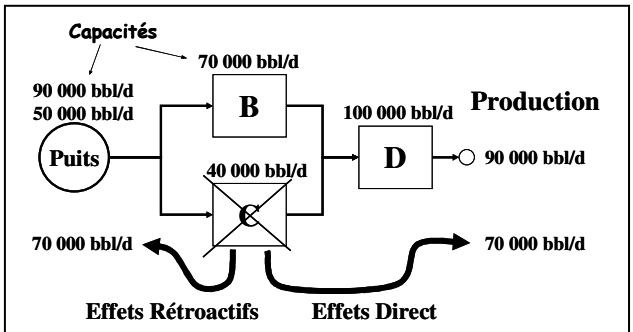


Figure 3 : Bloc diagramme Multi Etats

La figure ci-dessus illustre quelques unes des difficultés rencontrées.

Tout d'abord, et bien que cette figure ressemble comme deux gouttes d'eaux à un *Bloc Diagramme de Fiabilité (BDF)*, elle n'en est pas un car l'état de chaque bloc n'est pas binaire (marche / panne) mais est représenté par sa capacité de traitement qui peut varier en fonction d'événements internes (défaillances, réparations) mais aussi d'événements externes. Remarquer le puits qui possède deux niveaux de production.

A un instant donnée le niveau de sortie dépend des capacités actuelles des différents blocs et le niveau de sortie est imposé par les blocs qui imposent la production minimum (*goulot d'étranglement*). Il change donc en fonction de l'état des différents blocs :

- 90 000 bbl/d quand tout est parfait car c'est le puits qui impose la production ;
- 70 000bbl/d si le composant ***C*** est défaillant, car c'est B qui impose la production ;
- 0 bbl/d quand **D** est en panne car c'est lui qui impose la production
- ...

On identifie ainsi 5 niveaux de production à la sortie de notre petit système (0, 40 0000, 50 000, 70 000, 90 000) et c'est déjà beaucoup mieux que les BDF classiques qui n'en identifieraient que deux (marche / panne). Il suffira d'évaluer le temps passé dans chacun de ces niveaux pour évaluer la disponibilité de production du système étudié.

Avec 5 niveaux de production on ne rencontre pas vraiment de problèmes de modélisation mais il n'en est pas de même si ce nombre augmente ou, pire, si les niveaux ne sont pas identifiables a priori à cause de l' explosion combinatoire entraînée par un trop grand nombre de configurations possibles.

Dans l'analyse ci-dessus seul les effet "*aval*"des goulots d'étranglement sont pris en considération. Bien entendu pour les système réels il y a aussi des effets "*amont*" car la production étant assurée, in fine, par les puits, le niveau de production de ceux-ci puits doit être adapté en conséquence. Cet effet rétroactif peut avoir une incidence sur les calculs pour plusieurs raisons :

- Lorsque **D** est en panne, **B**, ***C*** ainsi que le puits vont être arrêtés et ne vont donc pas pouvoir tomber en panne pendant ce temps là ;
- Lorsque le composant ***C*** est défaillant, la production du puits sera ramenée à 70 000 bbl afin de pas "rejeter" d'huile à la mer !.
- ...

L'identification des contributions aux pertes de production peut aussi nécessiter la prise en compte de ces effets "amont".

D'autre difficultés de modélisation vont apparaître et on peut citer pêle-mêle :

- Les systèmes en attente (redondance froide) ;
- les défaillances de cause communes ;
- Les défaillances induites (la perte d'un système induit un état de production dégradée chez un autre) ;
- La mobilisation des supports d'intervention pour les réparations sous-marines ;
- La politique de maintenance curative (nombre d'équipes, défaillances *critiques* / *dégradées,* FIFO, ...)
- La politique de maintenance préventive (différée si la production n'est pas de 100%, par campagne où toute la production est arrêtée, ...)
- La politique d'approvisionnement des pièces de rechange ;
- Les délais non exponentiels des événements à prendre en compte ;
- Les délais de transport (hélicoptère, bateau, ..) ;
- Le rythme jour / nuit (suspension des réparations la nuit par exemple) ;
- La présence normale ou non de personnel à bord ;
- Les stockages ;
- Les conditions météo océanologiques ;
- Les dépôts de paraffines dans les conduites ;
- La formation des bouchons d'hydrate de carbone ;
- ...

Saut Qualitatif

Pour faire face aux difficultés ci-dessus la première idée qui vient à l'esprit est de penser aux bons vieux processus de Markov. Hélas un calcul rapide montre que, sauf cas particuliers, ils sont pratiquement hors sujet :

- Avec notre ancienne version de MARK EXD nous pouvions traiter une centaine d'états c'est à dire, une fois les agrégations pertinentes réalisées, un système comportent 6 à 7 composants ;
- Avec notre nouvelle version MARK XPR nous pouvons traiter quelques millions d'états c'est à dire, une fois les agrégations pertinentes réalisées, un système comportent 10 à 12 composants.

Autrement dit le passage de 100 à 1 000 000 d'état ne résout rien car il ne change pas franchement la taille des systèmes que l'on peut appréhender.

Pour traiter des systèmes de taille industrielle un saut *qualitatif* est nécessaire et il nous faut donc trouver autre chose ...!

Heureusement, la seconde idée qui vient à l'esprit est de mettre en oeuvre une simulation de *Monte Carlo* car dans une telle simulation seuls les états ayant quelque chances de se produire se manifestent effectivement.

Contrairement à l'approche analytique markovienne une telle approche est donc transparente à la taille du système étudié. Contrairement aussi à l'approche analytique markovienne, pas besoins de se torturer les méninges pour identifier les états négligeables afin de réaliser des approximation : ceux-ci ne se manifesteront tout bonnement pas !.

La méthode de calcul étant trouvée, reste à identifier un modèle pouvant lui servir efficacement de support en étant apte à modéliser correctement :

- Le fonctionnement ;
- Les dysfonctionnements ;
- La maintenance (cf. les difficultés plus haut) ;
- Les activités se déroulant en parallèle ;
- Les niveaux de production ;

- Les reconfigurations ;
- etc.

... bien entendu tout en restant très rapide au niveau des calculs !.

Au début des années 80 après avoir étudié les méthodes et outils disponibles sur le marché nous avons décidé de développer notre propre outils à partir des *réseaux de Petri Stochastiques interprétés*. Le logiciel **MOCA-RP** a été développé sur cette base et la première version date de 1982. Depuis le logiciel a fait l'objet de nombreuses améliorations. Lui aussi est encapsulé dans notre interface graphique interactive **GRIF** pour en faciliter l'utilisation.

Dès l'origine, l'algorithmique a été très étudiée afin de rendre les calculs le plus rapide possible. Ceci est en effet fondamental pour la simulation de Monte Carlo qui a la réputation (infondée en ce qui concerne les calculs de productivité) d'être gourmande en temps de calcul. Récemment une refonte complète en langage C a été réalisée en 2001 (**MOCA-RP V10**) pour améliorer encore ses performances et conserver sa portabilité.

L'ergonomie de l'interface **GRIF** a aussi été très soignée afin d'en rendre la mise en oeuvre simple et conviviale. Un simulateur pas à pas (*stepper*) permet à l'utilisateur de réaliser des simulations à la main et ceci est extrêmement efficace pour déverminer les modèles en cours de développement. **GRIF** a été réécrit en langage JAVA afin de lui assurer une grande portabilité.

Nous n'avons eu qu'à nous louer de ces choix puisqu'il nous a permis de faire évoluer le logiciel tranquillement en fonction des besoins au cours du temps, de l'exploiter à la fois sous WINDOWS et UNIX et de réaliser correctement toutes les études qui nous ont été demandées.

Les travaux réalisés à l'aide ce cet outil et notamment ceux portant sur la modélisation de la disponibilité de production ont fait l'objets de nombreuses communication dans des congrès nationaux comme le λμ ou internationaux comme ESREL.

Outils Nouveaux

Au début des années 2000 cependant, devant la complexité des systèmes à traiter et des questions qui nous étaient posées nous nous sommes posé la question de l'amélioration de ce modèle.

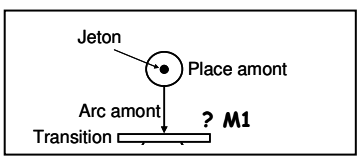
Deux voies s'ouvraient a nous et nous les avons mise en oeuvre toutes les deux afin d'en tester les performances :

- Réseaux de Petri stochastiques à "*prédicats*" ;
- Simulation stochastique basée sur une modélisation en langage AltaRica Data Flow.

La première voie est en parfaite continuité avec les modélisations que utilisons déjà depuis plus de 20 ans.

La seconde est nouvelle et basée sur un langage "*formel*" AltaRica développé au cours d'une thèse financée dans le cadre d'un projet conjoint entre plusieurs industriels français. Il a été spécialement conçu pour être capable de modéliser de manière détaillée le fonctionnement et les dysfonctionnement des systèmes industriels. La variante AltaRica Data Flow a été développé pour correspondre parfaitement à la description des systèmes de production.

Réseaux de Petri stochastiques à prédicats(RdPSP)



La figure n°4 ci-contre synthétise brièvement les divers éléments d'un RdPS "classique" :

- Places
- Arcs
- Transitions
- Jetons
- Messages

On y voit par exemple apparaître le message **?M1** reçu par la transition et qui permet de la valider lorsqu'il est dans l'état "**vrai**" ainsi que le message **!M2** qui est émis par la transition dans l'état "**vrai**" lorsque celle ci est tirée.

Ces messages qui sont utilisés pour faire communiquer entre eux des sous réseaux de Petri sont en fait la forme minimaliste des *prédicats* (**?M1**) d'une part et des *assertions* (**!M2**) d'autre part.

Dans la nouvelle version (**MOCA-RP V12**), toute expression mathématique dont on peut dire si elle est vraie ou fausse peut jouer le rôle de prédicat comme par exemple :
?? (#3 >=2), -M1, (A = 2*C)

Pour que la transition ayant ces prédicat comme "*garde*" soit valide il faut que le marquage de la place 3 (**#3**) soit supérieur ou égal à 2, que l'état du message **-M1** soit "**vrai**" (c'est à dire que celui de **M1** soit "**faux**") **ET** que **A** soit égal à 2 fois **C**.

De même, lorsque une transition est tirée, elle peut "*affecter*" de nouvelles valeurs à certaines variables comme par exemple :
!! B = 2 * #4, M3 = faux, D = (A + B) /2

Après le tir de la transition, **B** est égal à 2 fois le marquage de la place 4 (**#4**), le message **M3** est devenu **faux** et **D** est maintenant égal à la demi somme de **A+B**.

D'autre part, cette nouvelle version permet d'introduire des formules non attachées à des transitions particulières comme par exemple :
X = f(A, D, ...)
Y = g(A, M3, ...) où **f(.)** et **g(.)** représentent des expressions mathématiques quelconques.

Lorsque **A, D, ...** changent de valeur suite au tir d'une transition quelconque, les variables **X** et **Y** sont automatiquement recalculée pour être mises à jour.

Même si les changements entre l'ancienne et la nouvelle version apparaissent minimes, le pouvoir d'expression a été énormément augmenté et permet de simplifier énormément la modélisation de certains comportement difficile a représenter avec les RdPS classiques.

C'est le cas, en particulier pour les modèles de disponibilité de production.

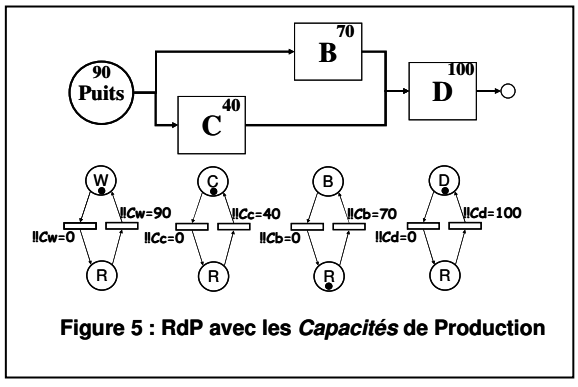


Figure 5 : RdP avec les *Capacités* de Production

Chaque bloc est représenté par un petit RdP à deux places (Marche / Réparation) et chaque fois q'il tombe en panne sa capacité de production est mise à 0 et elle est remise à sa valeur nominale lorsqu'il est réparé

Ainsi la production de ce système est représenté par la formule très simple suivante :
Prod = inf (Cw , Cb + Cc, Cd)

Cette simple formule donne en permanence la production du système et remplace à la fois tout l'ensemble de sous réseaux de Petri et de calculs annexes qui seraient nécessaire avec des RdPS classiques (pour détecter les niveaux de production, décrire l'architecture, ...)

On entrevoit sur cette figure 5 une manière de passer automatiquement d'un modèle BDF classique à un RdPSP permettant de réaliser des calculs de productivité. Des travaux sont en cours pour ce faire et à terme le module intégré dans **GRIF**, dénommé provisoirement "**StochaBloc**" (Bloc-Diagramme stochastiques), permettra de générer directement des modèles de productions en RdPSP à partir de description du type BDF "améliorés". Un prototype développé en 2003 à partir des simple RdPS donne déjà des résultats très prometteurs pour les calculs de fiabilité / disponibilité.

MOCA-RP V12 a été encapsulé dans GRIF courant 2003 de manière minimale. Des travaux sont en cours pour développer l'éditeur syntaxique et le simulateur pas à pas qui seront disponible fin 2004. Ces deux éléments sont indispensable pour une utilisation industrielle de cet outil. Cependant les essais de faisabilité réalisés depuis mi 2003 entrepris sont très encourageants. Bien entendu la vitesse de calcul est un peu plus lente que pour la modélisation classique mais les modèles étant plus légers et plus rapidement développés, l'un dans l'autre, cela semble s'équilibrer.

AltaRica Data Flow

Le langage AltaRica est un langage dit "*formel*" et à ce titre parfaitement défini syntaxiquement et sémantiquement parlant. Ceci lui confère des propriétés mathématiques rigoureuses qui permettent de transposer de manière automatique un modèle écrit en AltaRica dans un autre formalisme. Il s'agit d'une opération du même type qu'une "*compilation*" informatique.

Le langage AltaRica a été publié il y a quelques années pour être mis à disposition de la communauté des fiabilistes et des automaticiens. Un certain nombre d'applications en sont issues comme l'atelier OCAS-AltaRica développé par Dassault Aviation pour générer automatiquement des "*Arbres de Défaillances*".

Pour faciliter la modélisation des systèmes de production nous y avons introduit la notion de "flot de données" orientés et cela a donné la variante **AltaRica Data Flow** dont il est question dans cette communication.

Au sein de l'atelier ACACIA et en fonction des caractéristiques des modèles décrits en AltaRica Data Flow il est possible de générer automatiquement des arbres de défaillances, des séquences conduisant à un état de panne et des graphes de Markov. C'est ce dernier module qui utilise pour les calculs le logiciel MARK XPR dont nous avons parlé plus haut.

A part l'approche markovienne qui, comme cela a été montré auparavant, peut être utilisée dans certain cas simples, les autres modèles ne sont pas adaptés et c'est pourquoi l'atelier a été muni du *simulateur stochastique* "**SimStock**" permettant de réaliser des simulations de Monte Carlo directement sur un modèle écrit en AltaRica data flow. Chronologiquement il a été disponible avant MOCA-RP V12 et nous l'avons mis en oeuvre dès fin 2002 afin d'en tester les capacités.

Pour l'instant la saisie du modèle se réalise sous forme textuelle mais il est pourvu d'un simulateur pas à pas (*stepper*) indispensable pour déverminer les modèles en cours de mise au point. Des travaux sont prévu pour adapter l'interface de saisie graphique de l'atelier OCAS d'ici fin 2004 afin de faciliter l'utilisation de ce langage par les ingénieurs.

Quoi qu'il en soit, ce simulateur stochastique nous a permis de tester l'idée simple de modélisation des système de production que nous avions en tête depuis pas mal de temps. Elle consiste à considérer qu'un système de production est formé des éléments suivants :

- Les "*sources*" qui fournissent le produit brut à traiter (pour nous les puits de pétrole ou de gaz)
- les "*modules de traitements*" qui effectuent les divers traitement nécessaires sur le produit brut (pour nous les séparateurs, les compresseurs, etc.) ;
- Les "*récepteurs*" qui réceptionnent le produit fini (pour nous les stockages, les pipes d'expédition, etc.).

A un instant donné, sources, récepteurs et modules de traitement possèdent chacun un certaine "*capacité*" de production, réception et traitement. Ces capacité conditionnent la production observée à cet instant donné.

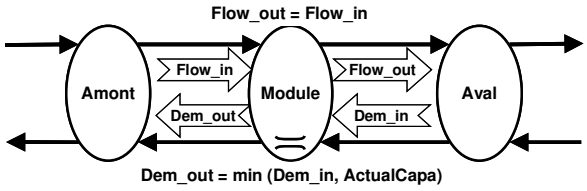
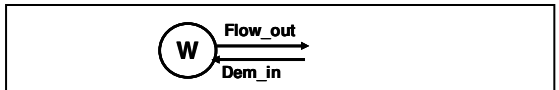


Figure 6 : Modèle de Production

La figure 6 montre le principe pour un module de traitement :

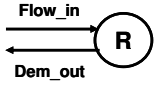
- Le module reçoit de l'aval (c'est à dire en provenance des récepteurs) une certaine "*demande*" de production ;
- Il retransmet cette demande vers l'amont (c'est à dire direction des sources) en fonction de sa capacité de traitement ;
- Cette demande se transmet de proche en proche jusqu'à atteindre une source qui y répond par un flot vers l'aval en fonction de sa propre capacité de production
- Le flot produit redescend de proche en proche et lorsqu'il atteint le module ci-dessus il sera intégralement retransmis vers l'aval. En effet, par construction du modèle le flot en provenance des sources est toujours inférieur ou égal à la capacité de traitement du module concerné.
- Finalement le flot qui arrive jusqu'aux récepteurs constitue la production du système.

Il est très facile de moduler les capacités des divers éléments en fonction des défaillances, réparations, re-configurations du système étudié. Le niveau de production se calcule ainsi automatiquement et il n'est plus nécessaire d'identifier a priori les différents niveaux possibles pour réaliser la modélisation.




```
node    Puits
flow    LocalCapa  :float :private;
        ActualCapa :float :out;
        Dem_in     :float :in;
        Flow_out   :float :out;
 assert  ActualCapa = LocalCapa,
        Flow_out   = min (ActualCapa,
Dem_in);
edon
```

Figure 7 : Modèle de Source




```
node    Recepteur
flow    LocalCapa  :float :private;
        ActualCapa :float :out;
        Dem_out    :float :out;
        Flow_in    :float :in;
assert  ActualCapa = LocalCapa,
        Dem_out    = ActualCapa;
edon
```

Figure 8 : Modèle de Récepteur

Les figure 7 et 8 montrent le modèle des sources et des récepteurs en langage AltaRica Data Flow.

Pour conserver la simplicité de ces modèles nous n'y avons pas introduit de défaillance. C'est ce que nous allons faire pour le module de traitement représenté sur la figure 9 ci-dessous.

```
node    Equipement
state   working :bool;
flow    LocalCapa  :float :private;
        ActualCapa :float :out;
        Dem_in     :float :in ;
        Dem_out    :float :out;
        Flow_in    :float :in ;
        Flow_out   :float :out;
event   Fail, Repair;
trans   working     |- Fail   -> working :=  false;
        not working |- Repair -> working :=  true;
init    working := true;
assert  ActualCapa = if working then LocalCapa else 0,
        Dem_out    = min (ActualCapa, Dem_in),
        Flow_out   = Flow_in;
extern  law < event Fail   > = exponential (Lfail);
        law < event Repair > = exponential (Mu);
        parameter Lfail      = 0.0001;
        parameter Mu         = 0.01;
edon
```

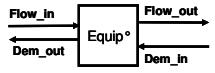
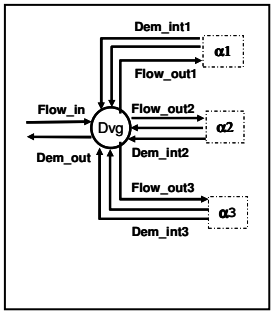


Figure 9 : Module de traitement

Dans ce modèle on voit apparaître les différents états du composant , les transitions qui y conduisent et le lois de probabilité qui y sont associées.



Bien entendu pour représenter un système de production il faut ajouter un certain nombre d'autres éléments comme les branchements par exemple.

La figure 10 présente un "*divergeant*" permettant répartir le flot vers l"aval et de regrouper les demandes vers l'amont.

Figure 10 : "Divergeant"

Les "*assertions*" correspondant à un tel divergeant sont les suivantes :

```
ActualCapa = α1 + α2 + α3,
Dem_out    = Dem_in1 + Dem_in2 + Dem_in3,
Flow_out1  = if (ActualCapa # 0)
             then Flow_in * a1/ActualCapa
             else 0,
Flow_out2  = if (ActualCapa # 0)
             then Flow_in * a2/ActualCapa
             else 0,
Flow_out3  = if (ActualCapa # 0)
             then Flow_in * a3/ActualCapa
             else 0;
```

Ce divergeant retransmet donc vers l'amont la somme des demandes et répartit les flots vers l'aval sont au prorata de coefficients $\alpha 1, \alpha 2$ et $\alpha 3$. Ces coefficients correspondent au capacités de traitement des branches qui lui sont connectées.

Un "*convergeant*" fonctionne exactement de la même manière mais en sens contraire. Il transmet vers l'aval la somme des flots en entrée et répartit vers l'amont les demandes au prorata de coefficients correspondant à la capacité des branches qui lui sont connectées.

Sources, récepteurs, modules de traitement et branchement forment les composant de base de la modélisation. Il reste à les connecter entre eux pour construire le système à étudier. Ceci est effectué dans un noeud spécial appelé "**main**" .

Ce noeud comporte différentes rubriques :

- **event** qui permet entre autres d'introduire de noms événements utilisés ensuite pour décrire des "synchronisations" (ex : **C1failure**) ;
- **Flow** qui permet d'introduire les nouvelles variables utile à la description du modèle (ex : **Production**) ;
- **Sub** qui permet d'énumérer les divers composants du système étudié (ex : **A:Equipement, W:Puits,** ...) ;
- **assert** sur laquelle nous reviendrons plus bas
- **sync** qui permet de décrire comment ertains événements doivent être synchronisés (ex **<C1failure:(C1.failure and F.pushC1)>** pour indiquer que lorsque C1 tombe en panne il prend sa place dans la file FIFO des réparations)
- **extern** pour introduire, en autre, les "*observateurs*" qui permettrons d'obtenir les résultats cherchés (ex: **observer production=weighted_value(<term E.output>**).

La rubrique **assert** est le coeur de la description car elle permet, de décrire toute la *topologie* du système étudié. Cela s'effectue très simplement par des formule du type **B.input=A.output, E.input=g.output, etc.** A l'heure actuelle cela est réalisé manuellement mais l'encapsulation en cours dans OCAS-AltaRica permettra, à teme de simplifier cette opération.

La rubrique **assert** permet aussi de décrire les paramètres du sytème (ex :**A.Capacity = 100**), les nouvelles variables nécessaires (ex: **α1, α2 , α3**), les états à observer , ...

Avec ces élément somme toute très simples il est possible de modéliser de manière relativement réaliste le comportement des systèmes de production.

Résultats

Stockages avec MOCA V12

Philippe Thomas

Cas test MINIPLANT

A RAUZY

Cas Test SAFERELNET

Marie BOITEAU

Conclusions

Les Réseaux de Petri Stochastique à Prédicats et le langage formel AltaRica Data Flow ont, du point de vue mathématique pratiquement le même pouvoir d'expression. Donc toute méthode développée pou l'un fonctionne avec l'autre. A l'heure actuelle les deux sont en test afin de mesure les avantages et inconvénients de chacun.

D'ores et déjà on a pu mesurer la puissance d'expression de ce type d'approche qui permet à l'analyste de modéliser à peu près tout ce qu'il veut. De ce point de vue on peut dire qu'ils préfigurent ce que devraient être les outils du futur.

Outre cet aspect trois autres points de vue primordiaux sont en prendre en compte pour assurer le succès de ces nouveaux outils :

- ergonomie des interfaces ;
- simulation pas à pas pour le déverminage ;
- vitesse des calculs.

Pour les cas test effectués avec les prototypes la vitesse de calcul s'est montrée tout à fait satisfaisante et c'est d'ailleurs pour cela éque nous avons persévéré dans leur développement. De plus des marges de progrès existent avec, en particulier la mise au point d'une "*machine virtuelle*" AltaRica data Flow en cours de réalisation et vers laquelle pourraient être compilés aussi bien les modèles AltaRica que les modèles RdPSP.

Les simulateurs pas à pas ne posent pas de problèmes. Celui d'AltaRica DF existe et celui des RdPS est en cours de développement.

Quant à l'ergonomie des interfaces elle sera celle d'OCAS pour AltaRica DF et celle de GRIF pour les RdPSP. Les utilisateurs desdites interfaces ls trouvent plutôt réussies.

Références

[1] Yves DUTUIT, Antoine RAUZY, Jean-Pierre SIGNORET et Philippe THOMAS. "*Modélisation d'un système dynamique et évaluation de sa sûreté de fonctionnement par réseaux de Petri stochastiques.* Acte λμ10 T2, pp. 648-659, 1996.

[2] E. CHATELET, Y. DUTUIT, J.P. SIGNORET et P. THOMAS. "*Dependability modelling and evaluation by using stochastic Petri nets : application to two test-cases"*. Reliability Engineering and System Safety, vol. 55, n° 2, pp. 117-124, 1997.

[3] Jean-Luc CHABOT, Tony HUTINET, Jean-Pierre SIGNORET. *"Comment cacher un réseau de Petri derrière un bloc diagramme de fiabilité"*, λμ13, Lyon 2002

[4] Yves DUTUIT, Jean-Pierre SIGNORET. *"Dynamic System Modelling by Using Stochastic Petri Nets and Monte Carlo Simulation"*, tutoriel présenté à Konbin03, Gdansk et ESREL 2003 Maastricht.

[5] Gérald POINT, "*AltaRica : Contribution à l'unification des méthodes formelles et de la sûreté de Fonctionnement"*, Thèse université de Bordeaux 1 2000

[6] G. POINT, P. THOMAS, S. LAJEUNESSE, A. RAUZY et J.P. SIGNORET. "*Le langage AltaRica"*. Actes λμ11, pp. 119-125, 1998.

[7] G. POINT and A. RAUZY. *"AltaRica – constraint automata as a description language"*. Journal Européen des Systèmes Automatisés, 33(8–9):1033–1052, 1999.

[8] A. ARNOLD, A. GRIFFAULT, A. RAUZY and. G. POINT. "*The altarica language and its semantics*". Fundamenta Informaticae, 34:109–124, 2000

[9] Antoine RAUZY. "*Mode automata and their compilation into fault trees*". Reliability Engineering and System Safety, Elsevier, vol. 78, pp 1-12, 2002.