Réseaux de Petri pour l'étude de la disponibilité opérationnelle des systèmes spatiaux en phases d'avant-projet

Jean-François Ereau

N° d'ordre : 2818 Année : 1997

Thèse

préparée au
Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

en vue de l'obtention du
Doctorat de l'Université Paul Sabatier de Toulouse

Spécialité : **Informatique Industrielle**

par
Jean-François EREAU

Réseaux de Petri pour l'Etude de la Disponibilité Opérationnelle des Systèmes Spatiaux en Phases d'Avant-Projet

---

Soutenue le 28 Novembre 1997, devant le jury :

Président	**Guy Juanole**
Rapporteurs	**Bernard Descotes-Genon**
	Yves Dutuit
Examinateurs	**Etienne Craye**
	Malecka Saleman
	Rodolphe Blondel
Directeurs de thèse	**Robert Valette**
	Hamid Demmou

Rapport LAAS N° 97465

A Dany, Jean, Claudie et Xavier

A Maurice

Remerciements

Une thèse est certes un travail personnel mais qui, sans le soutien, le conseil, la confiance et la complicité de certaines personnes serait une entreprise bien difficile, à l'aboutissement plus qu'incertain. Ceci est particulèrement vrai pour un travail de recherche appliquée comme celui qui est ici présenté. J'ai eu la chance de pouvoir compter sur l'appui de telles personnes que je tiens à remercier.

Tout d'abord je remercie les membres du jury qui m'ont fait l'honneur d'accepter cette responsabilité et de prendre connaissance de mon travail :
Guy Juanole, professeur de l'Université Paul Sabatier de Toulouse, président de ce jury, mon professeur à de nombreuses reprises et celui grâce à qui j'ai découvert les réseaux de Petri Stochastiques.
Bernard Descotes Genon, professeur à l'Université Joseph Fourier, Saint Martin d'Hères ainsi que *Yves Dutuit*, professeur à l'Université Bordeaux I, qui ont accepté la lourde tâche de rapporteurs.
Etienne Craye, professeur à l'Ecole Centrale de Lille, examinateur de ce travail.

L'idée de ce travail est née d'un noyau de personnes qui n'ont pas ménagé leur soutien depuis mon stage de DEA :
Merci à *Malecka Saleman*, ingénieur au CNES, « ma marraine » (que ce terme est juste !).
Merci à *Robert Valette*, directeur de recherches au LAAS-CNRS, un directeur que je souhaite à tous ceux qui veulent se lancer dans l'aventure d'une thèse.
Merci à *Hamid Demmou*, maître de conférence de l'Université Paul Sabatier.
Merci à *Albert Lehenaff*, chef du service *Qualité Etude*, et *Rodolphe Blondel*, ingénieur, d'ALCATEL ESPACE.
A ces personnes, je tiens à associer *Christpohe Lansade*, responsable de l'agence IXI Toulouse, qui m'a permis de profiter pleinement de ce contexte trés riche et passionnant du partenariat MISS-RdP.

Pendant presque la totalité de ma thèse j'ai travaillé au Centre Spatial de Toulouse du CNES. Je remercie donc *Christian Rouziès*, alors chef du département Sûreté de Fonctionnement et *Guy Hameury*, directeur de la direction Qualité, pour leur acceuil et les moyens matériels dont j'ai pu profiter. Merci également à *Laurent Raspaud* (qu'un jour je battrai au PACMAN !) et *Charles Lahorgues* qui ont réalisé le prototype logiciel. Je salue toute l'équipe Sûreté de Fonctionnement.

Le Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS fut, depuis le DEA, mon laboratoire d'accueil, j'en remercie ses directeurs *Alain Costes* et *Jean-Claude Laprie* ainsi que l'équipe Système de Production. Clin d'oeil à mon complice et ami *François Girault*.

Je salue les membres du service Qualité Etude et mes collègues et amis d'aujourd'hui du Département des Nouveaux Systèmes de Télécommunication d'ALCATEL ESPACE.

Enfin un merci tout particulier à ma femme *Réjane* qui, pendant toute cette période de rédaction particulièrement difficile et prise sur notre temps libre, m'a toujours soutenu.

Table des Matières

Introduction

Les études de Disponibilité Opérationnelle des systèmes spatiaux sont à la jonction des analyses systèmes et des études de Sûreté de Fonctionnement menées dès les toutes premières phases d'un projet. Si, dans le passé, elles étaient de portée limitée principalement en raison de la non maintenabilité des véhicules spatiaux, elles doivent répondre aujourd'hui à de nouveaux besoins.
Tout d'abord, les systèmes spatiaux se limitent maintenant rarement au seul segment spatial composé d'un unique satellite. En effet ils sont aujourd'hui fréquemment basés sur plusieurs satellites qui doivent s'interfacer avec des composantes sols et même à d'autres systèmes spatiaux déjà existants. Par ailleurs, le succès des services offerts par ces systèmes et l'utilisation commerciale ou étatique qui en est faite imposent des contraintes opérationnelles et budgétaires très fortes. Les très nombreux projets actuels de télécommunication qui sont basés sur des constellations de satellites illustrent pleinement cette évolution. Et les études de Disponibilité Opérationnelle, qui portent dès lors sur des systèmes fortement distribués, renouvelables, et dont la mise en service est complexe et progressive, doivent contribuer à leur juste dimensionnement. En effet, ce sont elles qui peuvent proposer des directions précieuses dans l'optimisation du couple coût / prise de risque.

Cependant, les fiabilistes en charge de la conduite de ces études sont confrontés à deux difficultés majeures. La première, d'ordre organisationnel, est liée à leur isolement certain des concepteurs système (décrié dans [Gory 92]) : ce type d'étude venant généralement de façon tardive dans un projet et n'impliquant que peu ces derniers. La deuxième est d'ordre technique, car les méthodes traditionnellement utilisées : Modèles Combinatoires (Arbres de Défaillances, Blocs Diagrammes de Fiabilité) ou Etats-Transitions (Chaînes de Markov entre autres) s'avèrent rapidement, comme on le verra, inadaptées.

L'utilisation des réseaux de Petri, dans un contexte d'Ingénierie Concourante, permet d'envisager des solutions séduisantes à ces difficultés. En effet, cette théorie conçue pour la description, l'étude et la commande des systèmes distribués et riche déjà de nombreuses années de recherche, propose un large éventail de résultats permettant de surmonter les limitations des approches plus classiques. On peut notamment citer :

- des règles de conception de modèles pour décrire des systèmes de taille importante,
- des possibilités de vérification formelle des modèles,
- de nombreuses extensions permettant de prendre en compte les phénomènes temporels et stochastiques inhérents aux études de Disponibilité Opérationnelle,
- des possibilités importantes de traitement par calcul analytique ou par simulation.

Par ailleurs, les règles d'évolution des réseaux de Petri sont très intuitives et très simples à comprendre par des non spécialistes. De plus, elles peuvent être clairement illustrées par un support graphique efficace qui favorise la démarche de conception en poussant chaque intervenant d'une équipe intégrée à se poser les bonnes questions pour parvenir à un accord sur la description du système.

Cependant, l'utilisation directe des réseaux de Petri pour les études de Disponibilité Opérationnelle des systèmes spatiaux, et dans un contexte industriel fortement contraint, n'est pas aisée. A cela plusieurs raisons. Tout d'abord, s'ils sont étudiés déjà depuis longtemps dans le domaine de la recherche, ils sont encore que peu appliqués dans le secteur industriel et il est donc difficile de se baser sur un retour d'expérience. Par ailleurs, si la compréhension de modèles déjà construits est aisée, leur conception nécessite un savoir faire qui ne peut être acquis sans une formation spécifique qu'encore très peu de fiabilistes ont suivie. Enfin, les spécificités inhérentes tant aux systèmes spatiaux qu'à ce type d'étude de Sûreté de Fonctionnement doivent être précisément étudiées afin d'isoler, parmi la foule de résultats théoriques sur les réseaux de Petri, leur domaine d'applicabilité.

Le travail présenté ici, et mené au sein des équipes Sûreté de Fonctionnement de l'agence française de l'Espace (CNES) et d'un industriel du spatial (Alcatel Espace), a pour objectif majeur de favoriser l'utilisation des réseaux de Petri pour ce type d'études.
Pour aborder de façon exhaustive toutes les spécificités de ce contexte, quatre directions ont été suivies :

- formation aux approches classiques d'étude de Disponibilité Opérationnelle,
- prise en charge d'études sur des projets spatiaux en cours de développement et pour lesquels les précédentes approches étaient mises en échec,
- étude bibliographique portant sur la théorie des réseaux de Petri (modèles logiques, extensions temporelles et stochastiques) ainsi que sur les approches orientées réseaux de Petri et portant sur l'évaluation des attributs de Sûreté de fonctionnement de systèmes industriels,
- participation active aux spécifications d'un simulateur de réseaux de Petri Temporels Stochastiques développé dans le cadre d'un partenariat industriel entre plusieurs acteurs du spatial et de l'aéronautique.

Sur la base de cette expérience et de façon concourante à ce qui précède, nous avons étudié les principes d'une approche d'aide à la modélisation intégrant les spécificités du domaine et que nous avons illustré par un prototype logiciel.

La première partie est dédiée au cadre général du travail réalisé.
Le chapitre I présente l'évolution du rôle des études de Disponibilité Opérationnelle après avoir souligné les spécificités des phases d'avant-projet des systèmes spatiaux.
Le chapitre II rappelle les principes des approches classiques (modèles combinatoires ou Etats-Transitions) tant pour la modélisation que pour l'évaluation. C'est l'occasion de mettre en évidence les limites face à la complexité du type d'études qui nous préoccupe.

La deuxième partie illustre les possibilités des réseaux de Petri face à ces limitations.
Le chapitre III et le chapitre IV, sur la base d'un exemple familier des fiabilistes : l'études des systèmes en redondance passive p parmi n, présentent les différents concepts utiles pour ce type d'étude : modélisation logique, vérification de modèles, prise en compte du temps et exploitation quantitative.
Le chapitre V met en oeuvre ces différents concepts sur une étude de cas : La Disponibilité Opérationnelle du Segment Spatial d'une Constellation de Satellites.

L'objet de la troisième partie de ce mémoire est de présenter une approche pour la génération automatique de réseaux décrivant la structure d'un système, comprise ici comme ses liens de dépendance du point de vue des défaillances, des fins de vie et des maintenances de ses éléments. Cette génération est obtenue à partir d'une description sous un formalisme bien connu des fiabilistes: les Arbres de Défaillances. Le modèle obtenu, basé sur les réseaux de Petri, a été baptisé *Arbre de Défaillances Dynamique*.

Le chapitre VI définit l'objet théorique servant de support aux *Arbres de Défaillances Dynamiques*. Il s'agit d'une composition fortement contrainte de réseaux de Petri Synchronisés [Moalla 78] [David 92] : les arborescences de réseaux de Petri à synchronisation structurée.

Le chapitre VII est consacré aux *Arbres de Défaillances Dynamiques*. On présente les principes de génération automatique, les règles d'évolution ainsi que la mise en oeuvre informatique que nous avons réalisée.

PREMIERE PARTIE

CADRE GENERAL

Chapitre I

Systèmes Spatiaux et

Disponibilité Opérationnelle

I.1. Systèmes Spatiaux

I.1.1. Enjeux et Spécificités

En un peu moins de quarante ans (premier vol Spoutnik en 1957), les activités spatiales ont connu un essor considérable. A l'origine de ce développement, la guerre froide bien sûr qui a fait de l'Espace son plus spectaculaire champ de bataille et peut-être le plus utile au bout du compte. Car les sommes « astronomiques » investies, pour démontrer la supériorité d'un système sur l'autre par la maîtrise de l'Espace, ont permis un effort de recherche sans précédent dans son intensité, sa diversité et ses enjeux. Et au-delà du prestige étatique ce sont bien de « nouvelles frontières » qui ont été révélées.

Communiquer, Observer, Se Déplacer, Fabriquer, Exploiter, telles sont les fonctions de l'homme qui ont pris une nouvelle dimension grâce à l'accès à l'Espace. En voici quelques déclinaisons.

Communiquer ...
Communiquer entre deux points fixes,
Communiquer entre un point fixe et l'ensemble d'une zone géographique,
Communiquer entre deux voire plusieurs points mobiles,
et ce, sans limites de distance, sans infrastructures de transmission terrestres et de façon massive.

Observer ...
Observer l'Univers, le climat, les ressources naturelles, le voisin, l'ennemi.
Observer pour comprendre, prédire, localiser, cartographier, surveiller, espionner, vérifier.

Se Déplacer ...
Se Déplacer autour de la Terre, dans sa proche banlieue, hors du système solaire.

Fabriquer ...
Fabriquer de nouveaux médicaments, de nouveaux matériaux.

Exploiter ...
Exploiter les ressources hors de notre planète.

On imagine sans peine les enjeux politiques, stratégiques, commerciaux, scientifiques et même philosophiques inhérents aux activités spatiales. Certains se nomment prestige, indépendance et sécurité des états, d'autres parts de marché ou encore protection et gestion des ressources naturelles, découverte et compréhension de l'univers, de son origine.

Face à de tels enjeux, si la réalisation expérimentale de systèmes spatiaux perdure, une véritable industrie s'est développée pour laquelle les états ne sont plus les seuls donneurs d'ordre. Cette industrie conserve toutefois des spécificités bien marquées liées aux contraintes de l'environnement spatial. En effet, l'Espace représente un environnement hostile pour l'homme et dont les conditions physiques diffèrent singulièrement de celles rencontrées sur Terre. Les radiations, les variations de température (-170 °C, +150 °C autour de la Terre), le vide, la micro-gravité, les vibrations et chocs liés aux lancements, les débris, les météorites, la difficulté de sortir de l'atmosphère, de la gravitation terrestre, d'y revenir également, sont autant de contraintes qui nécessitent la mise en œuvre de technologies complexes et innovantes.
L'industrie spatiale se caractérise donc par :

- un haut niveau de maîtrise technologique
- une multiplicité des métiers
- des possibilités de maintenance du véhicule spatial très réduites
- des coûts très élevés
- des productions peu récurrentes
- une évolution rapide liée à des applications de plus en plus nombreuses et variées
- des durées de réalisation et d'exploitation importantes
- un impact médiatique certain

I.1.2. Composantes

Un système spatial est constitué d'un ensemble de composantes hétérogènes qui interagissent de façon très variée. Quelles que soient les dénominations employées selon les agences [Pouzet 94], [Larson 92], on retrouve pour chaque mission les éléments suivants résumés par la Figure I.1 :

- *Le(s) Véhicule(s) Spatial(aux)*, encore appelé Segment Spatial. Ils diffèrent selon le type de mission (satellite, sonde, station orbitale, avion spatial, ...). La partie d'un véhicule spatial strictement utilisée pour la mission s'appelle la *Charge Utile* tandis

que la structure destinée à lui fournir les ressources nécessaires à son fonctionnement en vol se nomme la *Plate-forme*.

- *La Constellation Orbitale*. Elle caractérise les trajectoires du ou des véhicules spatiaux depuis le lancement jusqu'à la fin de vie.
- *Les Moyens de Commande/Contrôle*. Ils assurent la surveillance et le fonctionnement du segment spatial.
- *Les Moyens de Lancement*. Ils représentent le(s) centre(s) de lancement, le(s) véhicule(s) de lancement et les opérations associées jusqu'à la séparation entre le(s) lanceur(s) et le(s) véhicule(s) spatial(aux).

Ce premier ensemble de composantes caractérise le Secteur Spatial et fait appel à des sciences et techniques bien spécifiques à l'environnement spatial. Les autres composantes sont :

- *L'Objet de la Mission*. Directement lié à ce pour quoi la charge utile a été conçue, ce sera, par exemple, un type de relief en altimétrie, l'information à collecter en télécommunications ou encore la température atmosphérique en météorologie. Une même mission a souvent plusieurs objets.
- *Les Utilisateurs*. Ils correspondent dans tous les cas au maillon final de la chaîne fonctionnelle mais peuvent également être confondus avec l'objet de la mission comme dans le cas de la téléphonie mobile.
- *Le Segment Sol de Mission*. Il a pour mission de recevoir, traiter et diffuser les informations produites par le(s) charge(s) utile(s).

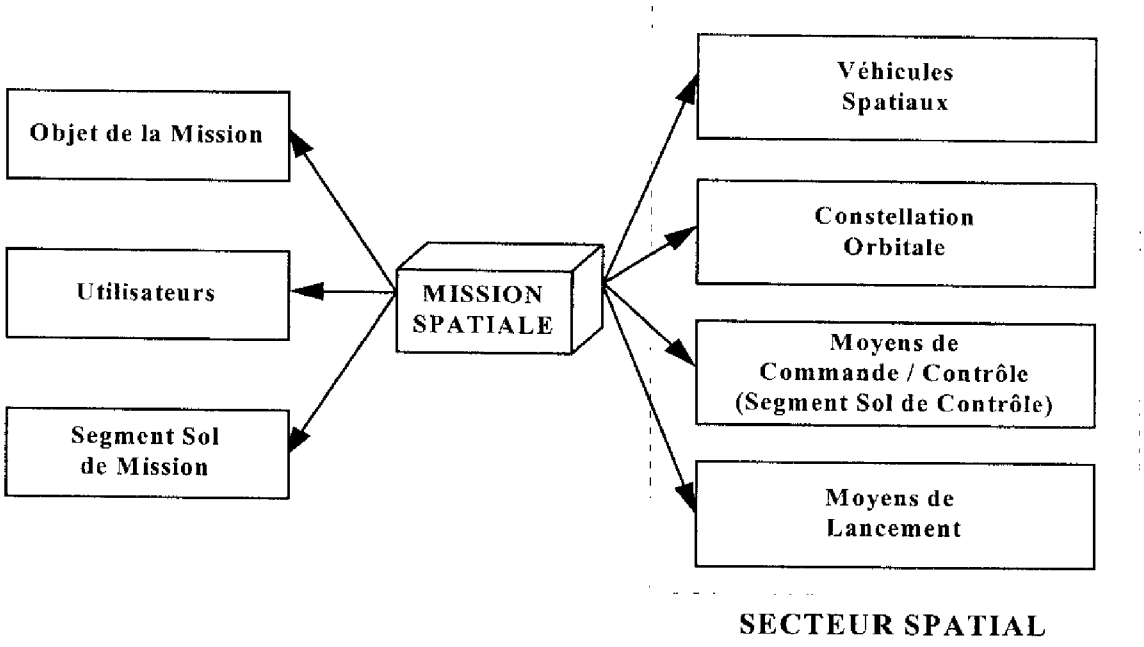


Figure I.1: Composantes d'un Système Spatial

Afin d'illustrer de façon sommaire ces différentes composantes, la Figure I.2 présente l'architecture générique d'un système de télécommunication mobile par satellites (particulièrement d'actualité). Le segment spatial est constitué d'une constellation de satellites à orbite basse. Les moyens de lancement correspondent à une série de tirs de lanceurs ayant à

leur bord plusieurs satellites pour le déploiement et le renouvellement du segment spatial. Ces satellites sont commandés et surveillés par des centres de contrôle/commande répartis autour de la terre afin de pouvoir intervenir à tout moment. Les centres de mission sont divisés en régions d'utilisation et, en coordination avec les centres de contrôle, reçoivent les données des charges utiles pour les transmettre soit par liaison terrestre, soit à nouveau par le segment spatial, aux utilisateurs. Ils ont également en charge la facturation auprès des utilisateurs du système.

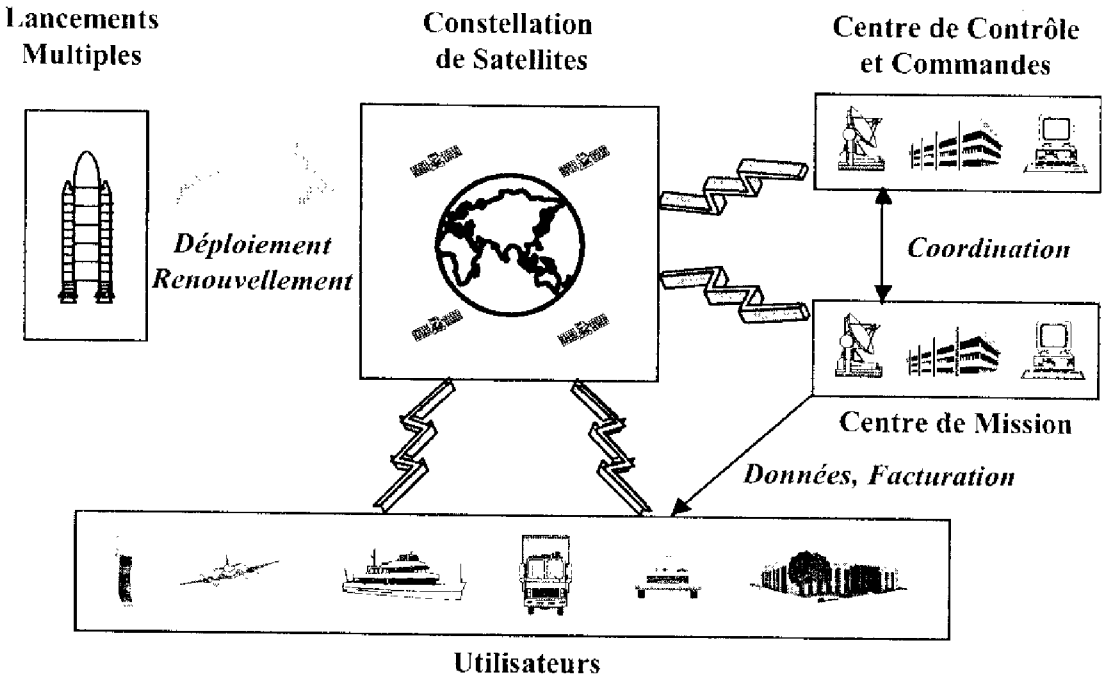


Figure I.2: Exemple de Composantes d'un Système de Télécommunication Mobile

I.1.3. Cycle de Vie

La conception, le développement et l'exploitation d'un système spatial nécessite une mobilisation très importante de moyens humains, matériels et financiers, et ce sur des périodes pouvant aller jusqu'à plusieurs dizaines d'années. C'est pourquoi, pour des raisons évidentes d'organisation et de coordination mais également de suivi et de vérification, le cycle de vie d'un système spatial fait l'objet de normalisation [Subias 94], [Larson 92]. Ce cycle, qui concerne chacune des composantes du système et leurs interfaces, suit traditionnellement une courbe en « V » pour laquelle chaque étape est clairement définie. Les premières phases de définition du besoin, d'analyse de mission, de définition préliminaire et de définition détaillée (phases 0, A, B et C) sont descendantes, tandis que les étapes de réalisation, d'exploitation et de retrait de service (phases D, E et F) sont ascendantes. La fin d'une phase est sanctionnée par une revue de fin de phase. Ces revues font intervenir l'ensemble des protagonistes du projet (commanditaires, développeurs, opérateurs et clients) qui décident du passage à la phase suivante, du prolongement de la phase courante ou voire même de l'arrêt du projet.
La Figure I.3 schématise ces différentes étapes du cycle de vie d'un système spatial.

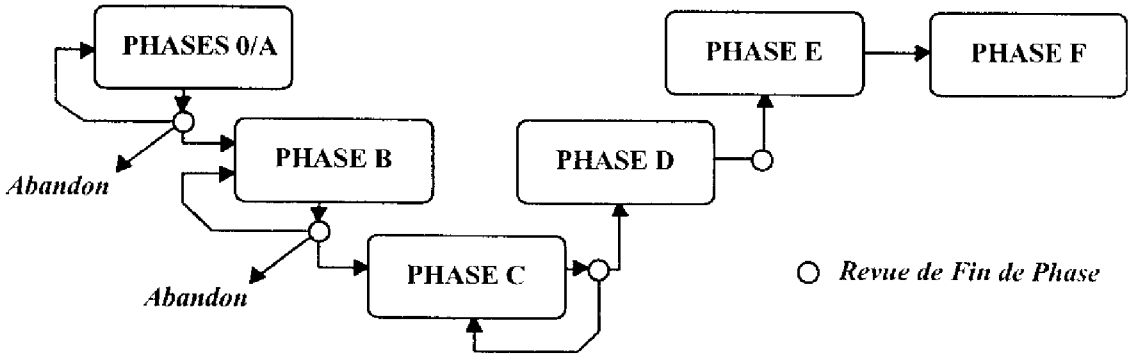


Figure I.3: Les Phases d'un Projet Spatial

I.1.4. « Better, Faster and Cheaper »

« Better, Faster and Cheaper » tels sont les leitmotivs actuels aussi bien des agences que des industriels du spatial. De quoi faire sourire bien d'autres secteurs industriels confrontés depuis longtemps déjà à ces objectifs incontournables, mais cela reflète « l'âge adulte » atteint par les activités spatiales. Comme on l'a vu, dans les premiers temps, ces activités étaient largement financées par la confrontation entre les deux blocs, et ce, en période de croissance économique. Les crédits militaires étaient très importants et les retours sur investissement se chiffraient plus en prestige politique et avance stratégique qu'en exploitation rentable de marchés. Aujourd'hui cette confrontation n'est plus d'actualité, les budgets militaires se sont largement étiolés et les acteurs du spatial se sont multipliés. De plus, avec le nombre considérable d'applications tant scientifiques que commerciales révélées, la notion « d'Espace Utile » prédomine. Conséquence inévitable, agences et industriels doivent intégrer fortement les notions de compétitivité et de productivité. Il ne s'agit plus de montrer au monde qu'on sait faire mais montrer à des clients qu'on sait faire de façon performante, à coûts modérés et dans les meilleurs délais.

Mais ce secteur bénéficie d'atouts importants pour assumer une telle tendance.
Atouts techniques tout d'abord grâce, notamment, aux progrès substantiels des sciences de l'informatique qui sont amplement pratiquées dans ce domaine de haute technologie. L'utilisation intensive de modèles informatiques a permis d'aider considérablement à la conception des systèmes spatiaux. Ces modèles peuvent en effet être simulés, testés et modifiés à volonté, l'expérimentation « réelle » ne venant qu'une fois ces derniers validés. Par ailleurs, l'informatique temps réel a permis de doter d'intelligence les véhicules spatiaux, maintenant souvent à même de vérifier leur bon fonctionnement, de diagnostiquer des pannes, de se protéger de leurs effets et, dans certains cas, de se reconfigurer. L'informatique de communication contribue également à l'amélioration de l'efficacité du processus de développement d'un système spatial (nous y reviendrons par la suite). Les progrès de la micro-électronique enfin, ont permis de sensiblement diminuer la taille et la masse des composants (éléments essentiels dans la minimisation des coûts d'un véhicule spatial) tout en augmentant nettement leurs performances.
Ce secteur bénéficie également d'atouts humains et organisationnels. En effet, la démarche de conception de systèmes est actuellement sujette à de nombreuses évolutions, et ce, pour l'ensemble des secteurs industriels de haute technologie. Ces évolutions, regroupées sous le terme d'Ingénierie Concourante, ont pour but d'améliorer la productivité et la compétitivité en se basant sur une modification du processus de développement d'un produit ainsi que sur une

modification de l'organisation et du fonctionnement des équipes de concepteurs. Mais précisons cette notion grâce à la définition du « Concurrent Engineering » proposée par *the Institute of Defense Analyses* américain en 1986, et aujourd'hui largement acceptée.

> *« Concurrent Engineering is a systematic approach to the integrated, concurrent design of products and their related processes, including manufacture and support. This approach is intended to cause the developpers, from the outset, to consider all elements of the product life cycle from concept to disposal, including quality, cost, schedule, and user requirements. »*

Sans entrer dans une description exhaustive (pour une présentation détaillée se reporter à [Carter 92]), il s'agit, tout d'abord sur le plan organisationnel, de décloisonner les relations entre les différents protagonistes d'un projet en favorisant les échanges d'informations (notamment grâce au support informatique) et en allégeant les niveaux hiérarchiques entre les concepteurs. Ce n'est plus sur un chef de projet, cordonnant différents spécialistes, mais sur une équipe intégrée, que porte la responsabilité du bon déroulement d'un programme. Le produit n'est plus vu comme un assemblage de sous-produits développés isolément mais comme un système conçu par une « équipe système » pluridisciplinaire. Par ailleurs, aux relations client / fournisseur se substitue une relation d'étroit partenariat durant tout le développement afin de cerner au plus juste les besoins du client. Enfin, le processus de développement séquentiel classique laisse place à un processus beaucoup plus itératif, guidé par un retour d'expérience intensif, et pour lequel la prise en compte de l'ensemble des contraintes opérationnelles est réalisée au plus tôt, ce qui inclut naturellement la qualité, les performances et les coûts.

Une telle approche, combinée aux progrès technologiques, a déjà démontré son efficacité aux vues des objectifs précités mais sa mise en œuvre dans le spatial demande une véritable révolution culturelle chez les différents acteurs. Cependant, l'intérêt étant prouvé, agences aussi bien qu'industriels s'y préparent. On peut notamment citer le projet *S2000+ ou la Conception des Systèmes Spatiaux en l'An 2000* au CNES [Gory 92] dont le but est la mise en place d'un atelier de conception d'avant-projets fondé sur ces principes et pour lequel « l'aspect système » prédomine.

I.2. Sûreté de Fonctionnement des Systèmes Spatiaux

I.2.1. Historique

Les notions de Qualité et de Sûreté de Fonctionnement sont intimement liées. En effet, si on se réfère à [Villemeur 88], la Qualité est définie comme l'aptitude d'un produit ou d'un service à satisfaire les besoins des utilisateurs et la Sûreté de Fonctionnement comme l'aptitude d'une entité à satisfaire une ou plusieurs fonctions requises dans des conditions données. La Sûreté de Fonctionnement sera donc une des caractéristiques contribuant à la Qualité d'un système. Mais la différenciation entre ces deux concepts est récente, et, afin de bien cerner leurs rôles respectifs dans le domaine spatial, nous retraçons ici un bref historique de leur développement commun au travers de leur évolution au Centre National d'Etudes Spatiales (CNES) et plus particulièrement au Centre Spatial de Toulouse (CST).

La confiance que l'on peut avoir dans un système a été une problématique prise en compte très tôt dans les activités spatiales françaises (années 60). En effet, l'impossibilité de réparer en vol, le défi politique que constituait la venue de la France dans le cercle si restreint des nations lancées dans l'aventure spatiale et les déconvenues subies aux premiers essais, obligeaient à s'assurer que le véhicule spatial fonctionnerait une fois dans l'espace (ou qu'il pourrait l'atteindre !). Mais la démarche Qualité se bornait alors à la vérification, en phase de fabrication, des dispositifs électroniques embarqués.

Au cours des années 70-80 l'émergence du caractère opérationnel des systèmes spatiaux (MétéoSat par exemple) pose une nouvelle problématique: non seulement le véhicule spatial doit fonctionner une fois dans l'espace, mais il doit fonctionner sur toute sa durée de mission. C'est la notion de Fiabilité qui est alors mise en avant. Par ailleurs, le soucis de maîtriser des projets de plus en plus complexes, d'en suivre précisément chaque étape, de les formaliser, donne une nouvelle composante à la Qualité: la Qualité Projet. C'est ainsi qu'à la fin des années 70 la Qualité, désormais appelée Assurance Produit, regroupe trois activités:

- La Qualité Contrôle dont le but est de vérifier mais aussi de sélectionner les meilleurs dispositifs électroniques
- La Qualité Fiabilité qui s'attache essentiellement au calcul de la fiabilité d'un véhicule spatial à partir de celle de ses composants électroniques
- La Qualité Projet

Mais au CST ces activités ne sont pas encore clairement différenciées (fonctionnellement et géographiquement) des autres métiers du spatial, à savoir des spécialistes techniques et des responsables de la conduite des projets.

Le début des années 80 correspond à une véritable « crise des satellites ». C'est, en France, la perte du satellite Télécom 1B, la défaillance des enregistreurs magnétiques de SPOT 1, la perte de canaux sur TDF1. La direction centrale de la Qualité est alors créée au CNES en 1985. Deux ans plus tard, et après par le choc représenté par l'échec retentissant aux Etats-Unis de Challenger en 1986 qui frappe l'ensemble de la communauté du spatial, la Direction de l'Assurance Produit est mise en place au CST. Elle représente une centaine de personnes (mobilités internes et embauches importantes) et regroupe l'ensemble des métiers liés à la Qualité en une même entité. C'est désormais un service puissant, autonome, dont les

activités sont clairement définies et concernent les différents sous-systèmes d'un projet spatial: électroniques bien sûr mais également mécaniques et logiciels.
Dans le même temps, un mouvement général de développement des techniques d'analyse prévisionnelle de risques se développe fortement dans l'industrie non spatiale (nucléaire, ferroviaire, automobile, ...). Etude de la Maintenabilité, de la Sécurité et de la Disponibilité viennent s'ajouter à celle de la Fiabilité pour constituer une véritable discipline: la Sûreté de Fonctionnement. Les méthodes quantitatives progressent et les analyses qualitatives de risques se multiplient. Le colloque λμ de Strasbourg en 1988, organisé par le CNES, reflète l'émergence de cette discipline dans le secteur industriel. C'est l'occasion d'une prise de conscience collective de son importance dans le développement des systèmes modernes.
Ces études sont alors regroupées, à la Direction de l'Assurance Produit du CST, dans le département Sûreté de Fonctionnement. Son rôle concerne la gestion prévisionnelle des risques techniques pour les phases de conception d'un projet spatial (Phases 0, A, B et C) et passe par leur identification, leur évaluation, et leur réduction; les autres activités de la direction s'attachant plus à l'analyse (contrôle, expertise, sélection) des différents sous-systèmes et au suivi du déroulement d'un programme.

Une telle organisation de l'Assurance Produit a toujours cours au CST et se retrouve dans les différentes agences (ESA, NASA, ...) ainsi que chez les industriels du spatial. Toutefois, chez ces derniers, elle est naturellement plus développée pour les activités liées à la production qui ne concernent pas les agences spatiales.

I.2.2. Concepts

Comme on vient de le voir, la caractérisation de la Sûreté de Fonctionnement d'un système est aujourd'hui basée sur l'étude de quatre grandeurs fondamentales: la Fiabilité, la Maintenabilité, la Disponibilité et la Sécurité. Ces grandeurs font l'objet de définitions formelles et non équivoques que nous rappelons [Lapric 85] [Villemeur 88]. Par élément on entendra composant, équipement, sous-système ou système.

- La *Fiabilité* d'un élément est son aptitude à accomplir une fonction requise, dans des conditions données et pendant un intervalle de temps donné.
  Notée R(t), elle est mesurée par la probabilité que l'élément ne soit pas défaillant sur l'intervalle [0, t].

  R(t) = P(E est non défaillant sur [0, t])

- La *Maintenabilité* d'un élément est son aptitude à être réparé (maintenu) dans un intervalle de temps donné; c'est à dire à revenir dans un état dans lequel il peut accomplir une fonction requise, lorsque l'exploitation et la maintenance sont accomplies dans des conditions données, avec des procédures et des moyens prescrits.
  Notée M(t), elle est mesurée par la probabilité que l'opération de maintenance de l'élément E soit achevée au temps t, sachant que l'élément est défaillant au temps t = 0.

  M(t) = P(E est réparé sur [0, t])

- La *Disponibilité* d'un élément est son aptitude à accomplir une fonction requise, à un instant donné et dans des conditions données.

Notée A(t), elle est mesurée par la probabilité que l'élément E ne soit pas défaillant à l'instant donné.

$$\mathbf{A(t) = P(E \text{ non défaillant à } t)}$$

- La *Sécurité* d'un élément est son aptitude à éviter de faire apparaître, dans des conditions données, des événements critiques ou catastrophiques.

Dans le domaine spatial, on pourra, par exemple, s'intéresser à la fiabilité d'un satellite ou d'un de ses équipements, à la maintenabilité d'un centre de contrôle, à la sécurité d'une base de lancement ou d'un avion spatial, ou encore à la Disponibilité globale d'une constellation de satellites. De ces grandeurs fondamentales peuvent être déduites des estimations de durée liées aux événements de défaillance et de réparation d'un élément. On peut citer:

- Le MTTF (Mean Time To First failure): durée moyenne de bon fonctionnement avant la première défaillance,
- Le MTTR (Mean Time To Repair): durée moyenne de réparation,
- Le MDT (Mean Down Time): durée moyenne dans l'état défaillant,
- Le MUT (Mean Up Time): durée moyenne de bon fonctionnement,
- Le MTBF (Mean Time Between Failure): durée moyenne entre deux défaillances.

Bien d'autres attributs, selon les domaines d'application, permettent d'affiner la notion de Sûreté de Fonctionnement mais nous nous limiterons ici à ces concepts.

I.2.3. Démarche

La démarche de Sûreté de Fonctionnement d'un projet regroupe un ensemble d'études qualitatives et quantitatives. Les premières permettent d'identifier les risques et d'analyser leurs effets; les dernières donnent accès aux grandeurs préalablement définies. Ces études suivent les phases de conception, elles débutent donc par l'analyse du système global et de chacune de ses composantes majeures (ex: segment spatial) en phases 0/A, puis s'affinent progressivement pour produire, en phase de conception détaillée (phase C), une analyse précise des risques, du système global jusqu'au moindre de ses équipements. Une étude sera donc caractérisée par:

- La phase de développement à laquelle elle est appliquée,
- Son objet (station de contrôle, charge utile, carte processeur embarquée, ...),
- Sa nature (qualitative, quantitative),
- Son objectif (hiérarchisation de risques, modes de défaillance, calcul de fiabilité, ...),
- Ses limites.

De nombreuses études peuvent ainsi être menées tout au long du développement d'un programme. Elles suivent cependant toutes une même logique illustrée par la Figure I.4 [Villemeur 88], [CEI 91], [Rouziès 94]. C'est un processus itératif qui débute par le recueil d'informations pertinentes sur l'élément. A partir de ces dernières, une modélisation est réalisée permettant l'évaluation qualitative ou quantitative des risques. Une évaluation quantitative peut toutefois passer par une première phase qualitative. Les résultats sont alors

analysés et, s'ils satisfont les objectifs préalablement fixés, l'étude s'achève. Dans le cas contraire, on procède à une modification de l'élément qui peut être d'ordre très varié (ajout, suppression ou changement de répartition des composants, changement de stratégie de maintenance, ...): c'est la phase de réduction des risques. L'élément ainsi modifié est alors à nouveau étudié.

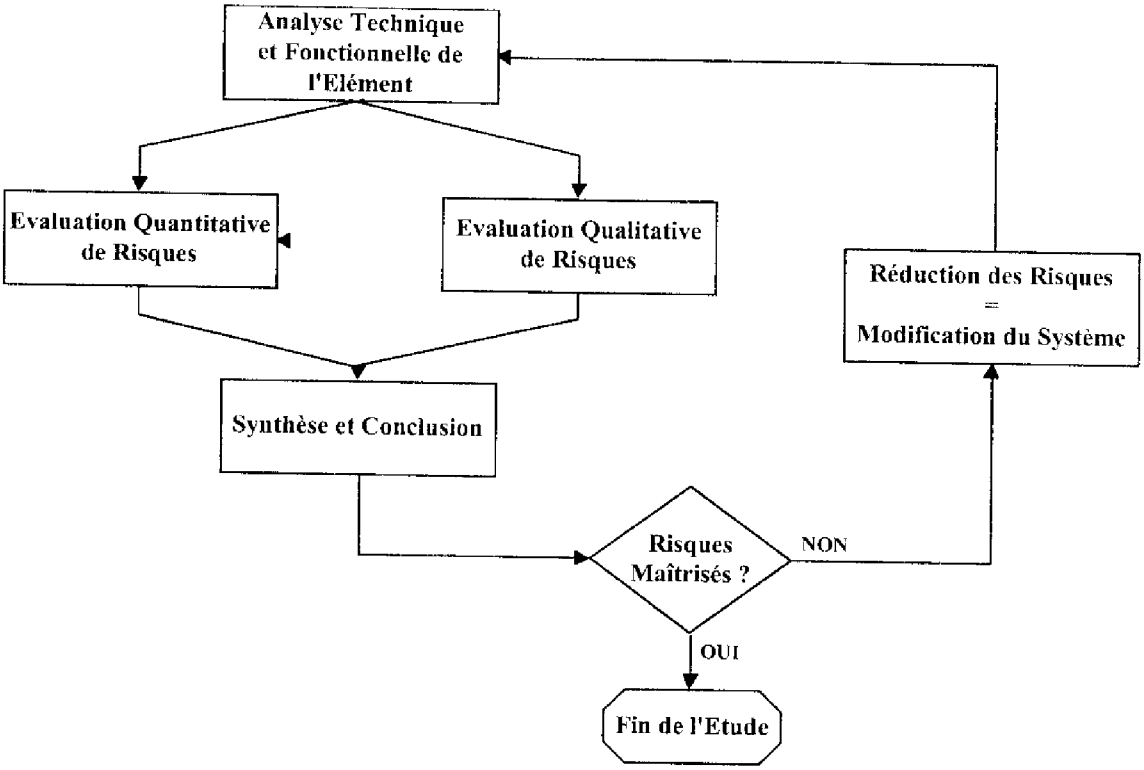


Figure I.4: Processus Itératif d'une Etude de Sûreté de Fonctionnement

I.2.4. Evolution

Après ce descriptif hors contexte, on pourrait penser que les études de Sûreté de Fonctionnement participent largement, et depuis longtemps, à la conception des systèmes. Dans le domaine spatial, elles ont été toutefois confrontées aux reproches formulés à l'encontre des activités Qualité. Lourdeurs supplémentaires, rôle d'audit mal perçu, données largement contestées, méthodes peu connues des concepteurs, telles sont les principales critiques. Si bien que dans les faits, elles étaient souvent réalisées en marge des équipes de conception et avec un décalage de temps sensible permettant rarement, et à fort prix, la modification d'un système.

Pourtant leur rôle est indéniable et va croissant avec le durcissement des contraintes opérationnelles des systèmes spatiaux. Non seulement elles contribuent à augmenter la confiance que l'on peut avoir dans un système, mais de plus, menées efficacement, elles permettent des économies substantielles dans le coût global d'un programme. Une étude américaine menée par le *Defense System Management College* en 1986 montre la répartition des coûts d'un programme spatial sur son cycle de vie dans les années 80 (cf. Figure I.5). On s'aperçoit que le coût lié à la maintenance et aux opérations est largement supérieur à celui du développement et de la production, cette disproportion étant essentiellement liée à une

mauvaise prise en compte, lors de la conception, des dysfonctionnements du système et des moyens de s'en protéger.

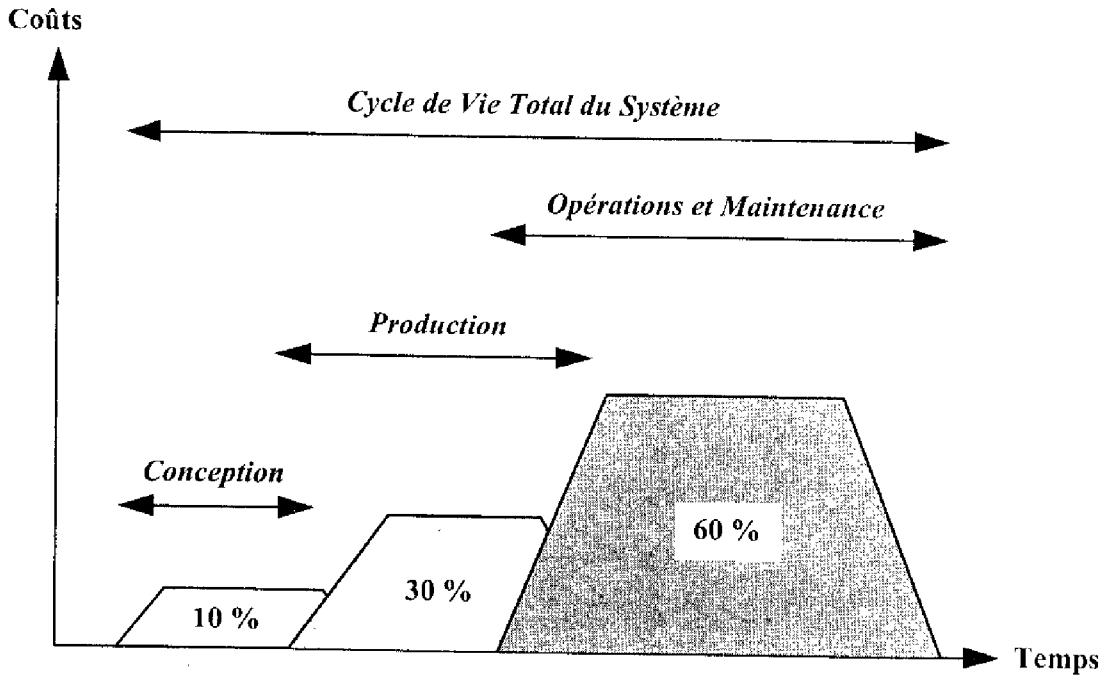


Figure I.5: Répartition des Coûts d'un Projet Spatial dans les Années 80

Des exemples similaires pourraient être trouvés dans, à peu près, tous les secteurs industriels ; c'est pour cela que, forte de constat, la démarche de l'Ingénierie Concourante cherche à intégrer, au plus tôt dans la conception, la prise en compte des risques et des coûts opérationnels. L'évolution en cours dans le spatial a donc pour but d'alléger et de simplifier la mise en œuvre des études de Sûreté de Fonctionnement mais surtout d'en changer la perception afin de les appliquer, voire de les intégrer, dès les études préliminaires de conception de système. Selon [Gory 92], la Sûreté de Fonctionnement doit être un véritable « design driver », les concepteurs doivent avoir présent à l'esprit, certes l'aspect fonctionnel et performance de leur système, mais également les possibilités de dysfonctionnement. De la vérification de la Sûreté de Fonctionnement, on doit passer à la construction de la Sûreté de Fonctionnement.

I.3. Disponibilité Opérationnelle, en Phases d'Avant-Projet, des Systèmes Spatiaux

Comme on a pu le souligner, l'Ingénierie Concourante vise à renforcer et à faire évoluer les phases d'Avant-Projet. Et parmi les fondements de cette démarche, la communication entre les concepteurs et avec le client, le développement des « analyses système » et la prise en compte des risques et des coûts sont des éléments essentiels. Nous allons illustrer ici comment les études de Disponibilité Opérationnelle en phases d'Avant-Projet des systèmes spatiaux contribuent à ces objectifs.

I.3.1. Avant-Projet

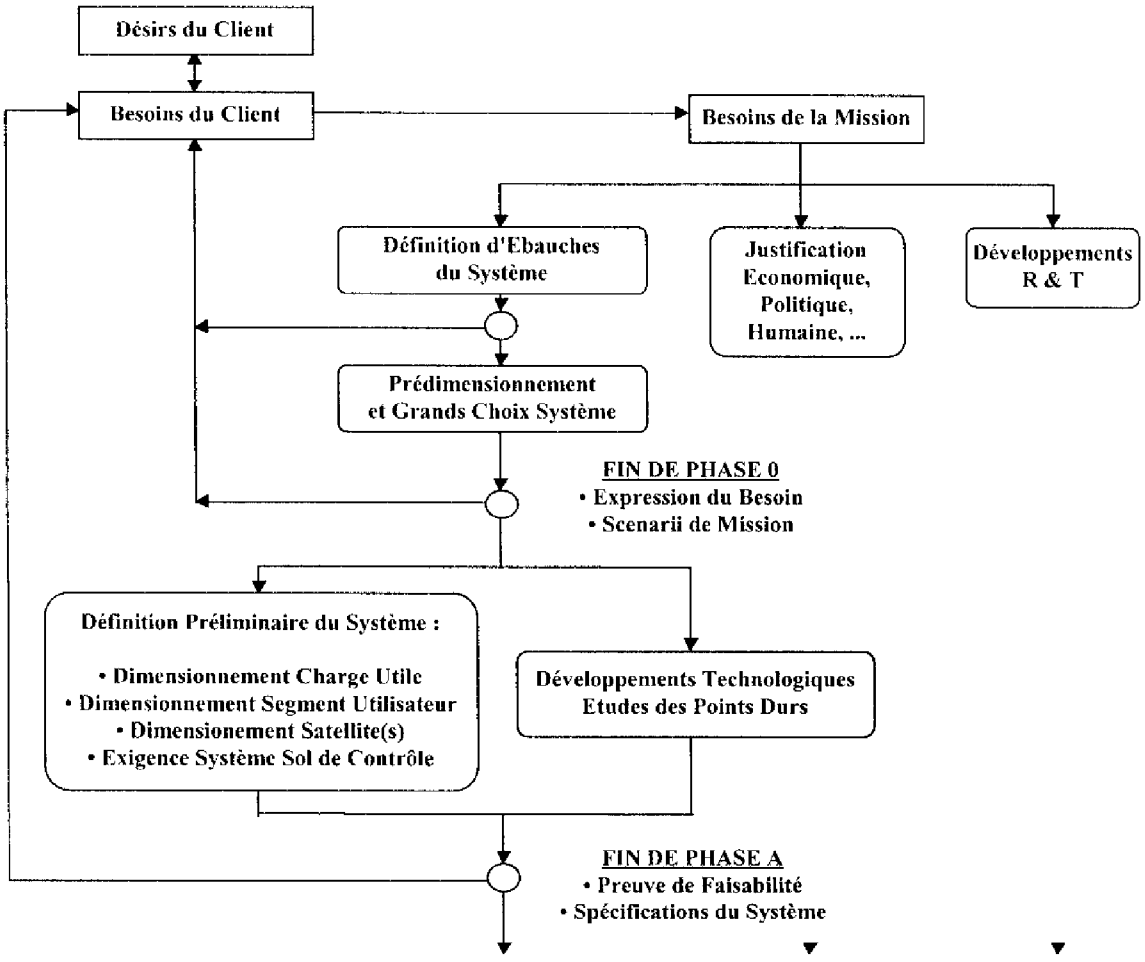


Figure I.6: Déroulement d'un Avant-Projet Spatial

S'il ne requière que peu de moyens financiers et humains, l'Avant-Projet est une étape tout à fait déterminante qui va largement conditionner la poursuite d'un projet spatial. Son objectif majeur est, à partir d'une expression précise des besoins du client, de prouver la faisabilité technique du projet en proposant une ou plusieurs solutions acceptables tant sur le plan technique que financier. C'est de la capacité à maîtriser cet objectif que les agences

spatiales tirent leur force de proposition et que les industriels emportent les contrats de développement.

L'Avant-Projet regroupe les deux premières phases du cycle de vie. Il concerne donc essentiellement l'aspect système et grands sous-systèmes. La Figure I.6 en décrit le déroulement. Le but de la phase 0 est, à partir des désirs du client et en étroite relation avec lui, de définir grossièrement des solutions possibles en termes de performances, coûts, délais et risques. La phase A approfondit les solutions envisagées en analysant l'impact sur le dimensionnement de chacune des composantes du système et en dégageant les problèmes techniques inhérents. Cette phase s'achève lorsqu'au moins une solution acceptable a été identifiée ou lorsque la faisabilité n'a pu être démontrée. Dans ce dernier cas le projet est abandonné, sinon la solution retenue permet de formuler les spécifications pour les phases ultérieures.

I.3.2. Etudes de Disponibilité Opérationnelle

Dans le passé, les études quantitatives de Sûreté de Fonctionnement des systèmes spatiaux se limitaient essentiellement à l'évaluation de la Fiabilité du véhicule spatial. A cela plusieurs raisons. Tout d'abord la fiabilité, comme on l'a vu, a été la première grandeur définie, les autres n'ayant complété le concept de Sûreté de Fonctionnement que bien plus tard. D'autre part, à cause de leur caractère irréparable, les composantes du système alors identifiées comme critiques étaient essentiellement le segment spatial et les moyens de lancement. Or, la plupart des missions se basaient sur un unique satellite généralement non reconfigurable suite à panne. Ainsi, les notions de Disponibilité du système global et de Fiabilité du satellite étaient confondues: la probabilité pour que le système fonctionne au cours du temps correspondait exactement à la probabilité pour que le satellite ne soit pas défaillant depuis sa mise en service.

L'émergence des contraintes opérationnelles des systèmes spatiaux et le progrès des techniques ont modifié considérablement cet état de faits. Et maintenant, pour faire face à certaines défaillances, le véhicule spatial est souvent doté de moyens de reconfiguration propres. Par exemple, un satellite pourra avoir deux émetteurs de télémesures (pour le segment sol de contrôle): un principal remplissant nominalement la fonction et un secondaire qui pourra prendre le relais en cas de défaillance du premier. Par ailleurs, pour une mission donnée, le segment spatial pourra être basé sur plusieurs véhicules spatiaux renouvelables comme dans le cas des constellations de satellites. Enfin, le

s segments sols de contrôle et de mission, en raison de la nécessité du service à assurer, seront également identifiés comme composantes critiques du point de vue des risques. Or ces sous-systèmes sont, eux, tout à fait réparables.

C'est pour l'ensemble de ces raisons que les études de Disponibilité des systèmes spatiaux se sont singulièrement différenciées des études de fiabilité de satellite. Elles sont devenues un élément important dans la faisabilité d'un système et contribuent ainsi largement aux objectifs des phases d'Avant-Projet, comme le montre cet extrait des recommandations officielles sur la Sûreté de Fonctionnement (en cours de rédaction) de l'Agence Spatiale Européenne [ESA 95] auprès des maîtres d'œuvre.

> *« The contractor shall perform availability analyses or simulation in order to assess the availability of the system. The results are used:*
>
> • *to optimize the system concept with respect to design, operations and maintenance,*
> • *to verify that the availability requirements are met,*
> • *to provide inputs to estimate the overall cost of operating system.*
>
> *a) The contractor shall perform the Outage Analysis in order to supply input data for Availability Analyses. The analysis output includes a list of all potential outages identified (as defined in the program), their causes, probabilities of occurrence and duration. Instead of outage probabilities, failure rates associated with outages may be provided. Furthermore, the mean of outage detection and the recovery methods shall be identified in the analysis.*
>
> *b) The Availability Prediction/Assessments shall be carried out at system level using the system reliability and maintainability models as well as the data from the Outage Analyses. »*

Elles constituent donc une réelle aide à la décision pour le dimensionnement de système. En effet, pour un scénario possible et un objectif de bon fonctionnement donné, elles permettent d'évaluer les ressources nécessaires à la vérification de cet objectif. Ainsi, différentes architectures mais également différentes politiques de mise en place, de renouvellement et de maintenance pourront être comparées permettant de dégager une solution optimale. En phase 0, ces études porteront sur plusieurs parties du système global, par exemple, le segment spatial, le segment sol (contrôle et mission) et les moyens de lancement; tandis qu'en phase A elles se concentreront plus exclusivement sur une des composantes comme le segment sol de contrôle (pour plus de précisions se rapporter à [Saleman 94]).

I.3.3. Données

En tant qu'études « système » réalisées très tôt lors de la conception, les évaluations de Disponibilité sont basées sur des données ayant deux caractéristiques essentielles: Disparité et Instabilité. Et, comme on le verra, toute la difficulté des démarches de modélisation et d'évaluation provient de ces particularités.

Disparité des données tout d'abord. Evaluer la probabilité pour qu'un système, sur lequel on peut agir, fonctionne correctement au cours du temps nécessite, en effet, de prendre en compte son architecture, les interactions entre ses différentes composantes, les risques de dysfonctionnement, mais également les moyens mis en place pour pallier les interruptions possibles de service. C'est ainsi que l'on peut regrouper les données d'entrée des études de Disponibilité selon trois grandes classes:

- *Les données « système »*. Elles concernent les différentes architectures possibles (nombre de satellites, de stations sols, ...), les contraintes d'utilisation (nombre de satellites nécessaires à la mission, nombre de pas de tir, durées de mise à poste, ...) ainsi que les durées des différentes phases de la mission.
- *Les données de risques*. Ce sont essentiellement les taux de défaillance estimés des différents sous-systèmes (courbes de fiabilité satellite, probabilité d'échec au lancement, à la mise à poste, ...).
- *Les données de soutien logistique*. Ces informations concernent tous les moyens et processus mis en œuvre pour prévenir ou pallier les défaillances possibles du système: niveau de redondance des différentes composantes (nombre de satellites supplémentaires par exemple), ressources matérielles et humaines (chaînes de production, capacités de stockages, ...) ainsi que les scénarios possibles de gestion pour la mise en place et la maintenance du système ou de l'une de ses composantes.

Ces données se différencient donc par leur provenance mais également par leur type. Elles peuvent être discrètes, temporelles, stochastiques ou encore correspondre à des processus logiques, comme ceux décrivant le remplacement d'un satellite, généralement peu formalisés et pas très clairs dans l'esprit des concepteurs.

Instabilité des données enfin. Instabilité liée à l'essence même des phases de conception préliminaire car à ce stade rien n'est figé, rien n'est précisément connu. Issues des besoins des utilisateurs, des choix et contraintes de conception, du retour d'expérience et des analyses de risques, ces données auront à être clarifiées, harmonisées et optimisées. C'est d'ailleurs un des buts des études de disponibilité, s'inscrivant en cela parfaitement dans les objectifs des Avant-Projets.

I.3.4. Démarche

En tant qu'études quantitatives de Sûreté de Fonctionnement, le déroulement des analyses de Disponibilité suit le processus général décrit par la Figure I.4. Mais, comme on vient de le voir, en raison de la disparité et de l'instabilité des données d'entrée, elles ne peuvent être réalisées isolément par un fiabiliste. La démarche menée s'appuiera donc sur une collaboration étroite avec les différents protagonistes d'un projet dans le contexte fortement itératif inhérent aux études de conception préliminaire. Cette équipe pluridisciplinaire procédera au recueil d'informations (tout à fait capital à la crédibilité de l'étude), à la validation des modèles conçus, à l'interprétation des résultats ainsi qu'aux modifications possibles des données d'entrée et des objectifs alloués. Le fiabiliste, pour sa part, animera ce travail collectif et sera en charge de la conception des modèles et de leur traitement quantitatif. Notons que l'on retrouve pleinement dans cette approche les principes de l'Ingénierie Concourante. La Figure I.7 en résume les étapes.

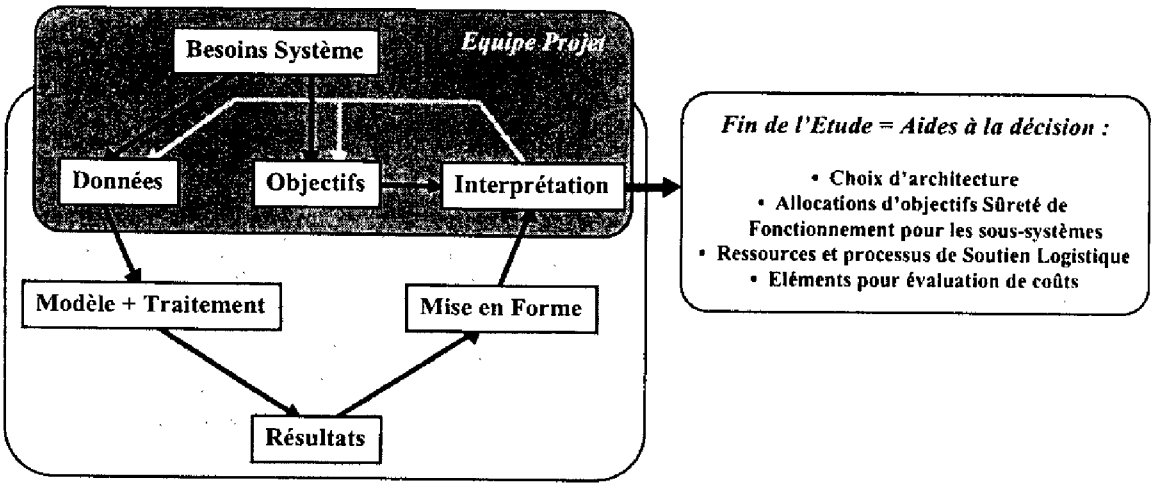


Figure I.7: Démarche d'une Etude de Disponibilité Opérationnelle

I.3.5. Conclusion

Les études de Disponibilité Opérationnelle des systèmes spatiaux ont gagné et en complexité au cours des dernières années. Elles doivent être prises en compte beaucoup plus tôt dans les phases de conception du système et les résultats qu'elles permettent d'obtenir sont tout à fait importants dans un contexte de maîtrise du couple risque/coût d'un projet.

Chapitre II

Méthodes « Classiques »

d'Etude de Disponibilité

Ce chapitre présente les méthodes classiques utilisées par les fiabilistes pour avoir accès quantitativement à la Disponibilité des systèmes ainsi qu'aux grandeurs dérivées. Notre but n'est bien évidemment pas de faire une présentation détaillée de toutes ces méthodes mais d'en relever les principes aussi bien pour la phase de modélisation des systèmes que pour la phase d'évaluation. Nous pourrons ainsi dégager un certain nombre de limitations de ces méthodes face auxquelles on est confronté pour les études de Disponibilité de Systèmes Spatiaux complexes. Si le lecteur désire avoir une description plus complète, il pourra se reporter à [Pagès 80] et [Villemeur 88].

Dans ce qui suit, nous supposerons que les systèmes étudiés sont cohérents.

Hypothèse II.1 **Systèmes Cohérents**

Un système est dit cohérent si:

- Lorsque le système est en panne, aucune défaillance ne rétablit le bon état du système,
- Lorsque le système est en marche, aucune réparation n'induit la panne du système,
- La panne de tous les éléments induit la panne du système,
- La bon fonctionnement de tous les éléments entraîne le bon fonctionnement du système.

□

Notations

On appellera

- *Composant* : un constituant terminal d'un système (n'ayant pas de descendance),
- *Macro-composant* : un constituant non terminal d'un système (ayant une descendance) et
- *Elément* : un composant ou macro-composant.

❑

II.1. Modélisation

II.1.1. Rappels: Modèles Dynamiques

Le processus de modélisation d'un système a pour but de réaliser une abstraction, appelée modèle, de la réalité de ce système. Cette abstraction n'est toujours que partiellement fidèle et ne décrit que les caractéristiques du système auxquelles on s'intéresse. Ces caractéristiques peuvent être statiques ou évoluer dans le temps: on parle alors de *modèle statique* (ex: plan d'architecte) ou de *modèle dynamique* (ex: équation de la trajectoire d'un mobile). Dans notre contexte on s'intéresse bien évidemment aux modèles dynamiques.

Les caractéristiques, ou variables, d'un modèle qui évoluent dans le temps définissent l'état du modèle (et par abus, du système) en fonction du temps. Selon que ces variables prennent leurs valeurs dans des ensembles continus ou discrets, on parle de *modèle à espace d'état continu ou discret*. Si les variables définissant l'état d'un modèle sont à la fois de nature continue et discrète on peut dire que le modèle est *à espace d'état hybride*.

De même, les dates d'évolution du modèle, c'est à dire les dates de changement d'état, peuvent prendre leurs valeurs dans des ensembles continus ou discrets. Le modèle correspondant est alors *à temps continu* ou *à temps discret*. Si ces dates prennent leurs valeurs dans des ensembles à la fois discrets et continus on peut dire que le modèle est *à temps hybride*.

Un modèle dynamique est donc caractérisé par son espace d'état et son espace des temps. Le Tableau II.1 présente la classification des modèles dynamiques en fonction de la nature de ces deux espaces. Dans le cas où ces espaces peuvent être à la fois continus et discrets on parle généralement de *modèle hybride*.

		ESPACE DES TEMPS	
		Continu	***Discret***
ESPACE D'ETAT	***Continu***	Modèles Continus	Modèles Discrétisés
	Discret	Modèles Discrets	Modèles à Evénements Discrets
	Hybrides	Modèles Hybrides	Modèles de « Haut Niveau »

Tableau II.1: Classification des Modèles Dynamiques

Hypothèse II.2

Les modèles de Disponibilité auxquels on s'intéresse sont des *modèles dynamiques à espace d'état discret.*

❑

Cette hypothèse est couramment formulée dans le cadre des études quantitatives de Sûreté de Fonctionnement. En effet, les états d'un système que l'on cherche à observer dépendent directement des états des composants du système et des états liés aux processus d'initialisation et de maintenance. Or, d'une part, les états de bon fonctionnement et de dysfonctionnement des composants sont finis voire même souvent booléens (marche ou panne) et, d'autre part, les processus d'initialisation et de maintenance sont considérés à un niveau d'abstraction tel qu'il peuvent aisément être représentés par un nombre fini d'états (par exemple la séquence d'états: production satellite, campagne lanceur et mise à poste).

Hypothèse II.3

Les modèles de Disponibilité auxquels on s'intéresse sont des *modèles dynamiques à temps continu.*

❑

Nous présentons dans ce qui suit les principaux modèles traditionnellement utilisés pour les études de Disponibilité.

II.1.2. Modèles Combinatoires

Les modèles combinatoires ont été définis pour étudier des problèmes de Fiabilité et/ou de Sécurité. Leur but est de décrire, sous forme graphique, les conditions logiques de bon fonctionnement ou de défaillance d'un système. Ils ne sont pas basés sur une représentation explicite des états mais décrivent, de façon non ambiguë, les liens de dépendance entre les éléments d'un système du point de vue de la Sûreté de Fonctionnement. Deux types de modèles sont utilisés par extension pour les études de disponibilité: *Les Arbres de Défaillances* (ou Arbres de Fautes) et *les Blocs Diagrammes de Fiabilité* (ou Diagrammes de Succès).

II.1.2.1 Les Arbres de Défaillances (AD)

Les Arbres de Défaillances décrivent les conditions logiques d'apparition d'un événement redouté en fonction de l'apparition d'événements indésirables. La notion d'événement doit être vue ici comme une variable booléenne passant à vrai et y restant à partir de l'occurrence d'un phénomène déclenchant, l'événement caractérise alors un changement d'état.

La logique d'apparition de l'événement redouté est représentée graphiquement par une structure arborescente (au sens de la théorie des graphes) et selon les principes suivants:

- La racine de l'arborescence représente l'événement redouté,

- Les feuilles représentent les événements indésirables de base,
- Les noeuds non feuilles et non racine sont des événements indésirables composés,
- Chaque événement non feuille est relié à ses fils par une porte logique *ET* ou *OU* décrivant les conditions d'apparition de cet événement en fonction des événements fils.

Les événements de base sont représentés par des cercles et tout autre événement par un rectangle. Les portes *ET* et *OU* conservent leur traditionnelle représentation graphique rappelée par la Figure II.1.

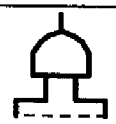
	***Porte OU***: L'événement du niveau supérieur est réalisé si l'un des événements du niveau inférieur l'est.
	***Porte ET***: L'événement du niveau supérieur est réalisé si tous les événements du niveau inférieur le sont.

Figure II.1: Portes de Base des Arbres de Défaillance

La description qui précède correspond à la définition des Arbres de Défaillances de base proposée dans les années 60 aux Etats-Unis pour l'étude du projet de missiles Minuteman. Nous restreignons ici notre présentation à ces Arbres de Défaillances « classiques ». De plus, on suppose que les événements pris en compte correspondent tous à des défaillances d'éléments d'un même système, l'événement redouté à observer est alors la défaillance du système. Ces Arbres de Défaillances sont donc directement liés à l'architecture matérielle du système et donnent graphiquement l'équation booléenne de la variable « *Défaillance Système* » en fonction des variables « *Défaillance Elément* ».

Notation

Soit E_i un élément, on notera $\overline{E_i}$ la variable booléenne « *Défaillance de* E_i ».

❑

Exemple II.1

Supposons un système spatial composé de deux satellites et d'une station sol. Le système est défaillant lorsque les deux satellites sont défaillants ou lorsque la station sol est en panne.

La représentation par Arbre de Défaillances correspondante est présentée par la Figure II.2. On note *S*, *SSol*, *SSpat* et Sat_i, respectivement le système, la station sol, le segment spatial et un satellite i, $i = \{1,2\}$. L'équation booléenne des variables « Défaillance » est dans ce cas triviale :

$$\overline{S} = \overline{SSol} + \overline{SSpat} = \overline{SSol} + \left(\overline{Sat_1} \times \overline{Sat_2}\right)$$

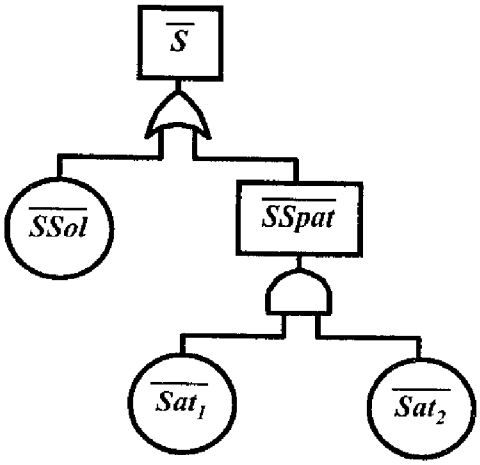


Figure II.2: Exemple Simple d'Arbre de Défaillances

❑

L'analyse qualitative des Arbres de Défaillances se base sur la détermination automatique des coupes minimales à partir de l'étude de la structure arborescente. Il s'agit, en fait, d'exprimer la variable « Défaillance Système » sous forme $\Sigma\Pi$ minimale, chaque monal représente alors une coupe minimale et correspond donc à une cause possible de la défaillance du système.

$$\overline{S} = \sum_{i=1}^{N} C_i \qquad \text{(II.1)}$$

avec $C_i = \prod_{j=1}^{N_i} \overline{E_j}$: coupe minimale et $\overline{E_j}$: défaillance d'un composant.

L'énumération automatique des coupes minimales est particulièrement intéressante dans le cadre des études de Sécurité car elle permet, sur des arbres très importants, de mettre en évidence les conjonctions de causes menant à un événement catastrophique.

II.1.2.2 Les Blocs Diagrammes de Fiabilité (BDF)

Les Blocs Diagrammes de Fiabilité sont une représentation duale de celle des Arbres de Défaillances: ils représentent les conditions logiques de bon fonctionnement d'un système en fonction du bon fonctionnement de ses composants. Ce sont des graphes orientés sans circuits ayant de deux sommets types: une source (ou entrée E) et un puits (ou sortie S). Les noeuds (ou blocs) de ces graphes correspondent à des éléments ou à des fonctions du système. Nous supposons ici que les blocs correspondent à des éléments matériels du système. Les blocs représentant des éléments dont la défaillance entraîne celle du système sont placés en série, ceux dont la défaillance n'entraîne la défaillance du système qu'en combinaison avec d'autres sont placés en parallèle. Chaque bloc peut correspondre lui-même à un sous-diagramme série ou parallèle. La Figure II.3 illustre les deux types de diagramme qui découlent de ces associations.

Notation

On confond le nom d'un élément E_i avec la variable booléenne « *Bon Fonctionnement de* E_i ». ❑

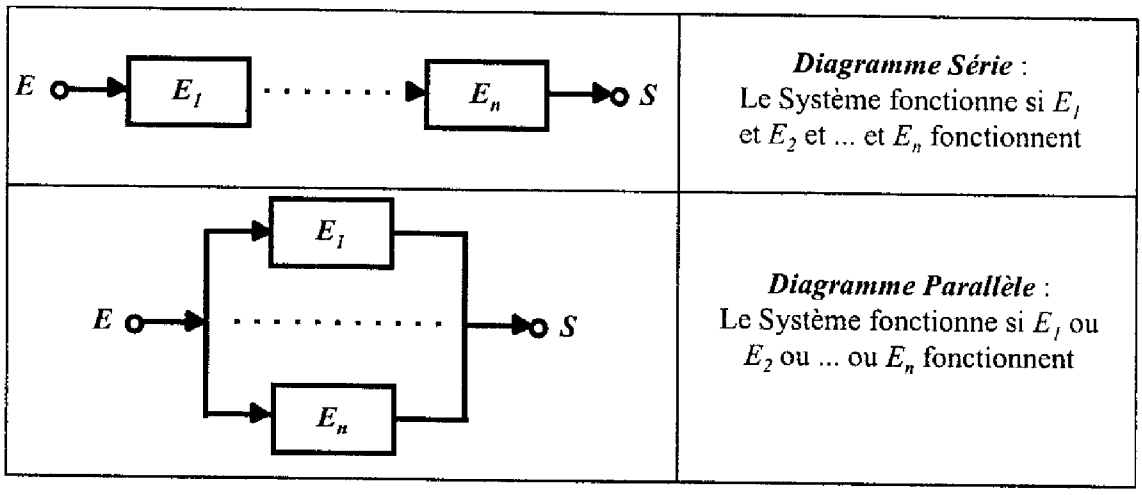


	Diagramme Série : Le Système fonctionne si E_1 et E_2 et ... et E_n fonctionnent
	Diagramme Parallèle : Le Système fonctionne si E_1 ou E_2 ou ... ou E_n fonctionnent

Figure II.3: Diagrammes Série et Parallèle

Les Blocs Diagrammes de Fiabilité représentent donc l'équation graphique de la variable booléenne « Bon Fonctionnement du Système » en fonction de celle de ses éléments.

Exemple II.1 **(suite)**

Si l'on reprend l'Exemple II.1, dans la mesure où l'on considère que chaque élément ne peut avoir que deux états (un état de bon fonctionnement et un état de panne), une autre façon de décrire les conditions logiques de défaillance consiste à dire que le système fonctionne si et seulement si la station sol fonctionne et au moins un des deux satellites fonctionne. La représentation par Blocs Diagramme de Fiabilité correspondante est présentée par la Figure II.4. L'équation booléenne du bon fonctionnement du système s'écrit alors:

$$S = SSol \times SSpat = SSol \times \left[Sat_1 + Sat_2\right]$$

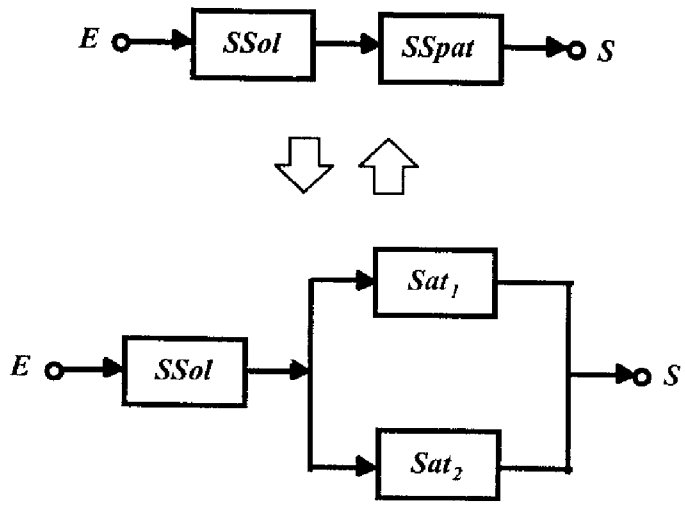


Figure II.4: Exemple Simple de Blocs Diagrammes de Fiabilité

❑

L'analyse qualitative des Blocs Diagrammes de Fiabilité se base sur la détermination automatique des chemins minimaux (ou liens minimaux) menant de E à S. Chaque chemin correspond à une combinaison d'éléments dont le bon fonctionnement est nécessaire pour assurer celui du système. L'énumération des chemins minimaux revient à exprimer la variable « *Bon Fonctionnement du Système* » sous forme $\Sigma\Pi$ minimale, chaque monal représente alors un lien minimal et correspond à une configuration possible et minimale de bon fonctionnement.

$$S = \sum_{i=1}^{N} L_i \qquad \textbf{(II.2)}$$

avec $L_i = \prod_{j=1}^{N_i} E_j$: chemin minimal et E_j : bon fonctionnement d'un composant E_j.

II.1.2.3 Arbres de Défaillances et Blocs Diagrammes de Fiabilité

Les Arbres de Défaillances décrivent les conditions qui mènent le système dans un certain état (l'état de défaillance) tandis que les Blocs Diagrammes de Fiabilité expriment, eux, les conditions selon lesquelles le système va rester dans un état donné (l'état de bon fonctionnement). Dans la mesure où l'on suppose, comme ci-dessus, que chaque composant ne peut avoir que deux états possibles, les Arbres de Défaillances et les Diagrammes de Fiabilité sont des représentations duales au sens de la logique booléenne. Les algorithmes de passage de l'une à l'autre de ces représentations peuvent être trouvés dans [Shooman 70] et [Malhotra 94]. La Figure II.5 présente les équivalences entre ces deux modèles.

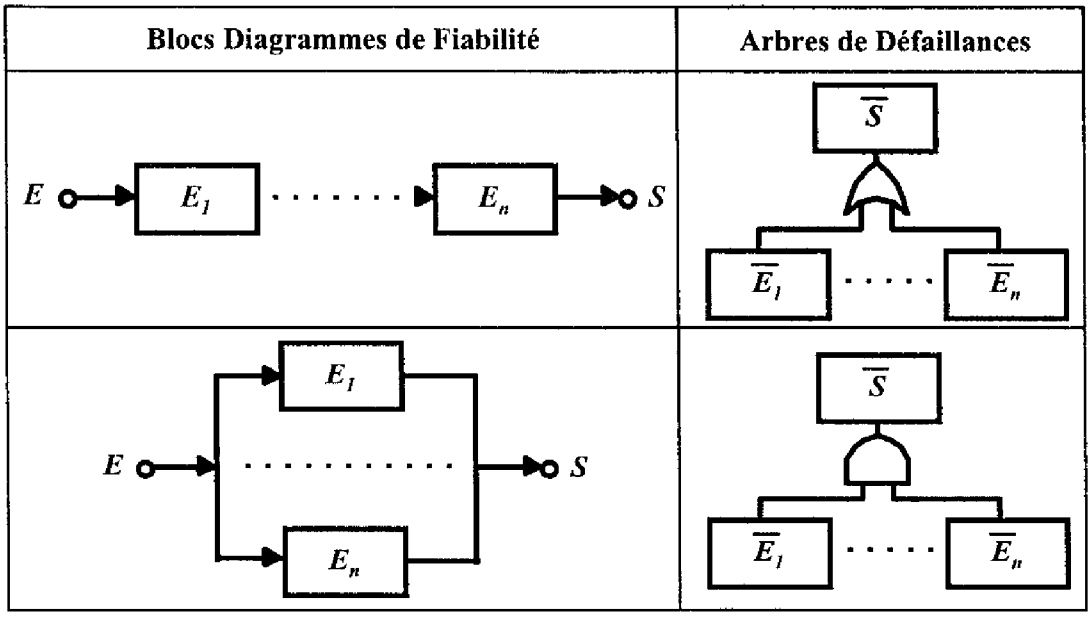


Figure II.5: Equivalences Blocs Diagrammes de Fiabilité et Arbres de Défaillances

***Exemple II.1* (suite)**

La Figure II.6 illustre les modèles AD et BDF équivalents décrivant les conditions de bon fonctionnement ou de défaillance du système spatial de l'Exemple II.1.

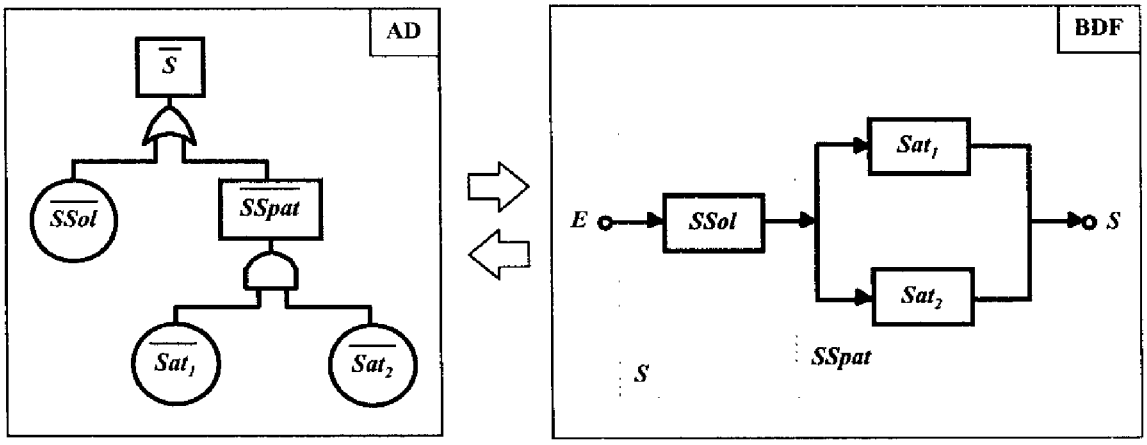


Figure II.6: Deux Modèles Possibles du Problème de l'Exemple II.1

❑

Les modèles combinatoires présentés jusqu'ici correspondent aux modèles de base initialement définis au début des activités de Sûreté de Fonctionnement. Ils caractérisent de façon très claire les liens de dépendance, du point de vue du bon fonctionnement (ou du dysfonctionnement), entre les éléments d'un système mais ils ne décrivent pas les conditions d'initialisation ou de maintenance de ces éléments. De plus, étant basés sur la logique booléenne et décrits par des structures arborescentes, leur pouvoir de modélisation est soumis à d'importantes restrictions. On peut notamment citer :

- L'impossibilité pour un élément d'intervenir plusieurs fois dans les conditions de bon fonctionnement (ou de dysfonctionnement),
- L'impossibilité de prendre en compte l'ordre de mise à vrai de variables,
- L'impossibilité de considérer, par élément, plus d'un état pertinent dans les conditions de bon fonctionnement (ou de dysfonctionnement),
- L'impossibilité de décrire des comportements dynamiques complexes.

Face à de telles limitations, de nombreuses extensions ont été définies au cours des dernières décennies. Nous en présentons ici quelques unes significatives.

La possibilité pour un élément d'intervenir plusieurs fois dans les conditions de bon fonctionnement (resp. de dysfonctionnement) a d'abord été considérée avec les diagrammes *redondance active p parmi n* (resp. *portes matricielles p parmi n*) des Blocs Diagrammes de Fiabilité (resp. Arbres de Défaillances). Un diagramme redondance active est un diagramme parallèle particulier pour lequel le bloc équivalent est en bon fonctionnement si au moins p blocs, parmi les n initialement actifs, sont non défaillants. De même une porte matricielle est une porte pour laquelle l'événement de niveau supérieur est réalisé si p parmi n événements fils le sont. La Figure II.7 montre les équivalences entre ces deux types d'association.

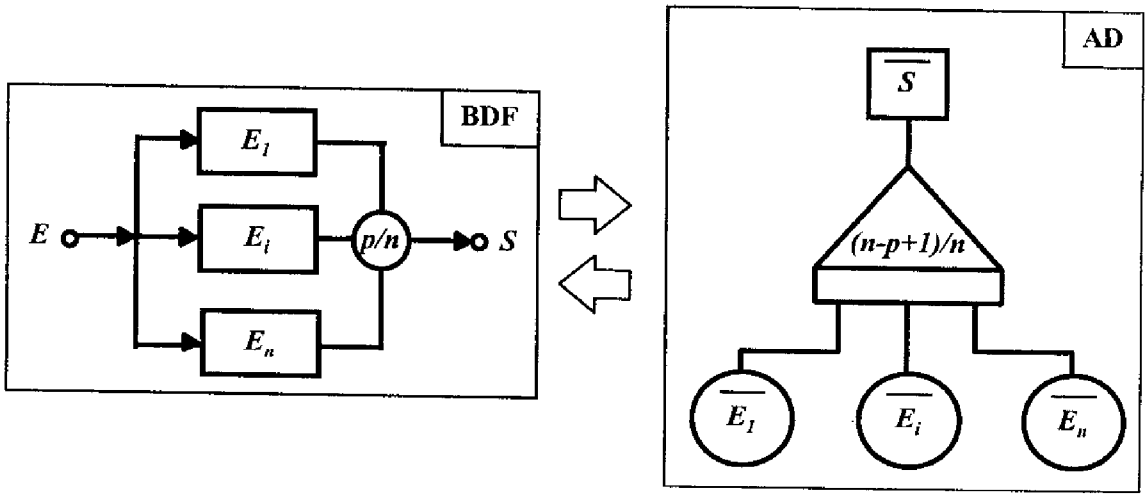


Figure II.7: Diagramme Redondance Active et Porte Matricielle

Les *Arbres de Défaillances à Evénements Répétés* (AD_ER) ou les *Blocs Diagrammes de Fiabilité à Blocs Répétés* (BDF_BR) [Buzacott 70] autorisent, de façon plus générale, un élément quelconque à être pris en compte à plusieurs niveaux. L'Exemple II.2 illustre cette possibilité.

Exemple II.2

On suppose un système spatial composé de deux satellites et de trois stations sols. Chacun des satellites est suivi par une station sol dédiée, la troisième station, peut suivre indifféremment chacun des deux satellites. Le système fonctionne si au moins un satellite fonctionne avec une station sol pour le surveiller. La modélisation des liens de dépendances entre ces éléments est donnée par l'AD_ER et le BDF_BR de la Figure II.8. On note Sat_i, Sol_i et *SOL*,

respectivement un satellite i, une station sol *i*, *i={1,2}* et la station sol commune. Les blocs (resp. événements) en gris correspondent aux blocs (resp. événements) partagés. Par soucis de concision, les événements intermédiaires de l'Arbre de Défaillances n'ont pas été représentés.

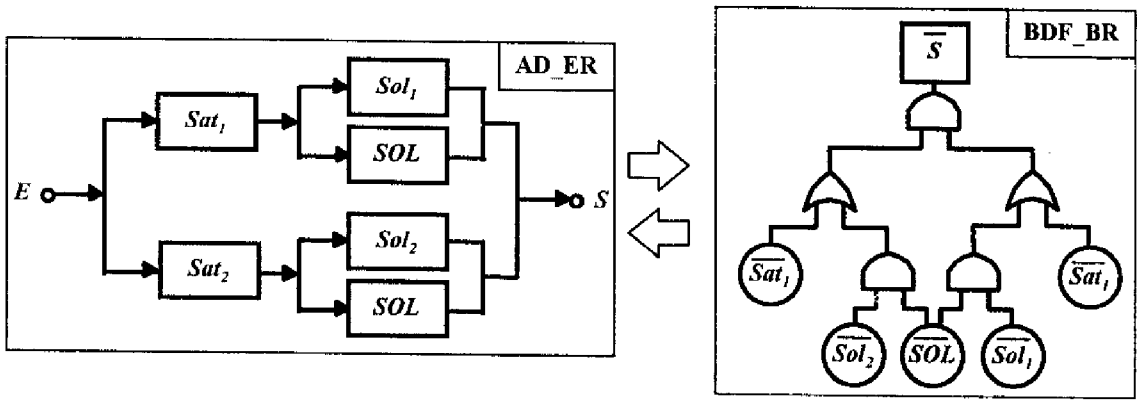


Figure II.8: Exemple de AD_ER et BDF_BR

❑

Les *portes redondances froides* [Dugan 92], [Dugan 96] permettent de représenter avec les Arbres de Défaillance des éléments qui peuvent avoir plus de deux états. La Figure II.9 présente une porte de ce type avec nos propres notations. Chaque noeud de l'arbre est alors à voir comme un élément matériel du système. Les éléments blancs sont supposés être initialement actif tandis que les éléments gris en attente. Le système est défaillant dés qu'il y a une défaillance d'un élément blanc. Dans ce cas un élément en attente peut prendre sa place, il échange alors sa couleur avec l'élément défaillant et le système redevient en bon fonctionnement.

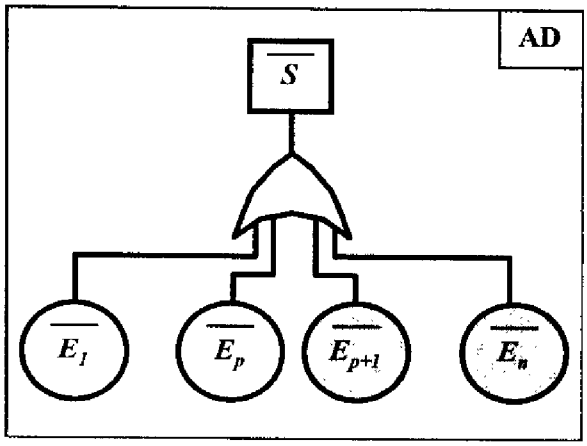


Figure II.9: Porte Redondance Passive

De telles extensions ne vont pas sans certaines imprécisions voire ambiguïtés car la logique sous-jacente de ces modèles combinatoires n'est plus respectée. Bien d'autres extensions ont été définies comme les portes séquentielles [Dugan 92], [Dugan 96] ou encore les diagrammes avec redondance passive [Villemeur 88]. Ces nouveaux modèles sont alors un support simple de communication entre les concepteurs, mais ils ne permettent pas

directement l'analyse qualitative ou quantitative. Des mécanismes de traduction automatique vers des modèles formels plus puissants sont alors nécessaires pour pouvoir les exploiter.

II.1.3. Modèles Etats-Transitions

Parmi les formalismes au pouvoir de modélisation plus étendu, les modèles Etats-Transitions (ou Machines à Etats) occupent une place importante. En effet, ces modèles ont soit le pouvoir d'expression des machines de Turing (pour un nombre d'états infini) soit celui des langages réguliers (pour un nombre d'états fini). Etant basés sur l'énumération complète des états et des transitions entre ces états, ils permettent, à priori, de décrire tous les systèmes à espace d'état discret. Dans la mesure où le nombre d'états est fini, on peut les représenter soit par des graphes soit par des matrices carrées. Dans une représentation par graphe, les sommets correspondent aux états du système et les arcs aux transitions entre ces états. Sous forme de matrice carrée, la dimension de la matrice est celle de l'espace d'états et chaque élément $t_{i,j}$ correspond à la transition de l'état *Xi* à l'état *Xj*. Dans le cadre des études de Disponibilité, on pourra faire une partition entre les états de bon fonctionnement et les états de défaillance.

La Figure II.10 présente, sous forme matricielle, le modèle générique Etats-Transitions pour les études de Disponibilité. On suppose ici qu'il y n_1 états de bon fonctionnement et n_2 états de défaillance. Les états de disponibilité du système sont les n_1 états non défaillants. Les transitions entre la classe des états défaillants et celle des états non défaillants correspondent aux sous-matrices B_{n_1,n_2} et C_{n_2,n_1}.

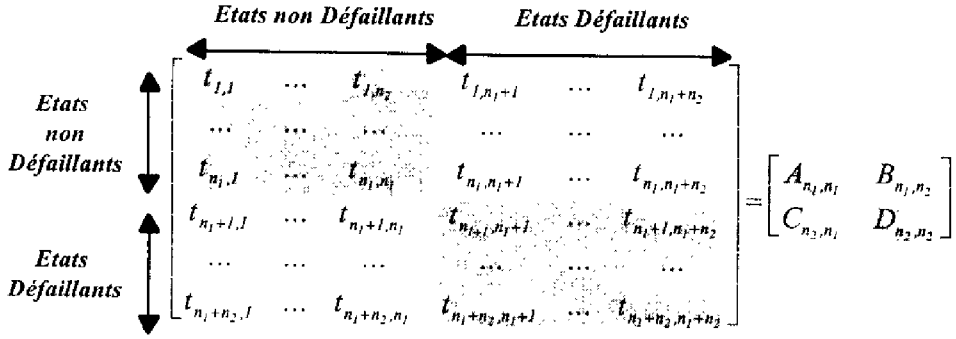


Figure II.10: Représentation Matricielle d'un Modèle Etats-Transitions de Disponibilité

L'analyse qualitative des modèles Etats-Transitions se base sur les très nombreux résultats de la théorie des graphes. On pourra par, exemple, déterminer les composantes fortement connexes du graphe sous-jacent pour mettre en évidence le caractère répétitif du modèle, étudier les chemins menant dans certains états critiques ou encore déterminer les états absorbants qui caractérisent les blocages du modèle.

Exemple II.3

On considère un système spatial composé de deux satellites Sat_1 et Sat_2. Ces deux satellites peuvent avoir *3* états: l'état actif (*ON*), l'état d'attente (*SB*) et l'état défaillant (*HS*). Un

satellite actif ou en attente peut défaillir. Initialement Sat_1 est actif et Sat_2 est en attente. Si Sat_1 défaille alors une commutation peut se produire de façon à ce que Sat_2 prenne sa place, c'est à dire devienne actif à son tour. Tout remplacement n'a lieu que lorsqu'il y a au moins un satellite défaillant et qu'il n'y a pas de commutation possible. Deux satellites ne peuvent jamais être actifs en même temps.

La Figure II.11 présente le modèle Etats-Transitions correspondant sachant que :

- *Don. i* correspond à la défaillance du satellite *i* actif,
- *Dsb. i* correspond à la défaillance du satellite *i* en attente,
- *Ron. i* correspond au remplacement en mode actif du satellite *i*,
- *Rsb. i* correspond au remplacement en mode attente du satellite *i*,
- *C. i* correspond à la commutation sur le satellite *i*,
- Chaque état du système est décrit par un vecteur colonne tel que:

$X_j = \begin{bmatrix} Etat\_Satellite\ 1 \\ Etat\_Satellite\ 2 \end{bmatrix}$ avec $Etat\_Satellite = \{OK, SB, HS\}$.

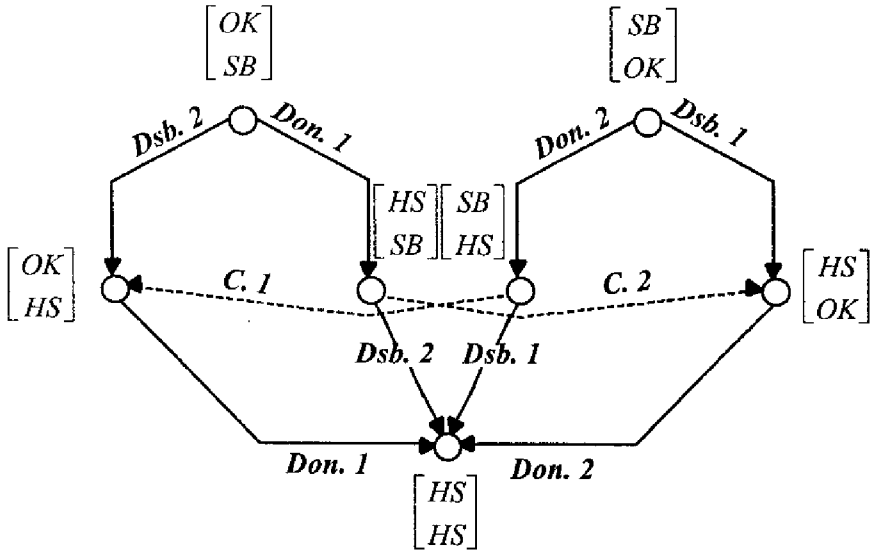


Figure II.11: Exemple de Modèle Etats-Transitions sous Forme de Graphe

Ce graphe est fortement connexe, donc il n'y a pas d'état absorbant, donc le système n'a aucun blocage et tous les états sont atteignables.

Les deux satellites ont un comportement complètement symétrique. Si on ne les distingue pas il est alors possible de replier le graphe de la Figure II.11 pour obtenir une vision agrégée de leur comportement.

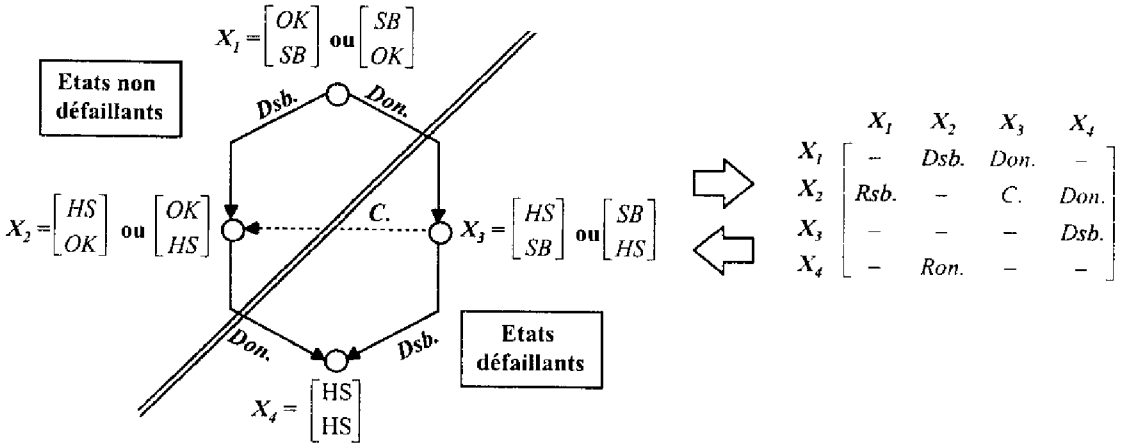


Figure II.12: Graphe Précédent Replié et Présentation Matricielle

❑

Dans la mesure où l'on ne s'intéresse qu'aux systèmes décrits par des modèles à espace d'état discret, c'est à dire ayant un nombre fini ou infini dénombrable d'états, on est en mesure de pouvoir représenter n'importe lequel de ces systèmes par des modèles Etats-Transitions.

Toutefois on est confronté à une limitation d'utilisation importante. Cette limitation est due à l'explosion combinatoire du nombre d'états d'un tel modèle. En effet, si un système est composé de N éléments ayant chacun deux états possibles (marche et panne), ce système aura 2^N états soit plus d'un million d'états pour seulement *20* éléments! Deux conséquences à ce problème d'explosion combinatoire. Tout d'abord la place mémoire (taille de la matrice des états) pour un traitement informatique sera rapidement exorbitante. Bien que des techniques de regroupement automatique d'états avant traitement existent, elles imposent l'énumération préalable de tous les états. D'autre part, la conception et la lecture de tels modèles sera fastidieuse et souvent source d'erreurs.

Nous n'avons décrit jusqu'à présent que certaines techniques de modélisation d'un système permettant de représenter sa logique d'évolution en fonction des événements et états liés aux défaillances de ses éléments. Nous abordons maintenant le problème du traitement quantitatif de ses modèles afin d'avoir accès au paramètre Disponibilité ainsi qu'à ces grandeurs dérivées.

II.2. Evaluation

II.2.1. Hypothèses Probabilistes

II.2.1.1 Processus Stochastique

Dans les études de Disponibilité auxquelles nous nous attachons, la nature aléatoire des systèmes étudiés provient exclusivement des dates d'occurrence des événements engendrant les changements d'état. L'évolution d'un modèle suit donc un processus stochastique à temps continu et à espace d'état discret.

Définition II.1 (Simplifiée) **Processus Stochastique**

Un *Processus Stochastique* $(X(t))_{t \in T}$ est défini par la donnée de deux ensembles:

- L'ensemble T des « instants » auxquels le processus est défini. Il peut être discret ou continu.
- L'espace d'état, X, formé des valeurs possibles d'une variable aléatoire $X(t)$, $t \in T$. Il peut être lui aussi discret ou continu. Pour les processus que l'on étudie, X est discret.

❑

Un processus stochastique à temps continu et à espace d'état discret (et fini) est donc caractérisé par la valeur du vecteur probabilité instantanée dont on rappelle la définition.

Définition II.2 **Vecteur Probabilité Instantanée**

Le *vecteur probabilité instantané* d'un processus stochastique à n états est le vecteur $(1 \times n)$:

$$P(t) = \left[P_{X_1}(t),\ P_{X_2}(t),\ldots,\ P_{X_n}(t)\right] \text{ tel que: } \begin{cases} 0 \le P_{X_i}(t) \le 1 \\ \sum_{i=1}^{n} P_{X_i}(t) = 1 \end{cases}, \ \forall i \in \{1,n\}$$

avec $P_{X_i}(t)$: probabilité d'être dans l'état X_i à l'instant t.

❑

Une propriété importante, qui peut simplifier singulièrement l'étude d'un processus stochastique, concerne la stationnarité de ce processus, c'est à dire sa capacité à atteindre un régime permanent. Pour un tel processus, un décalage arbitraire dans le temps ne modifie pas l'évaluation de la probabilité de ses différents états. Le calcul du vecteur probabilité stationnaire (noté Π) suffit souvent à caractériser les grandeurs espérées. Ce vecteur correspond en fait à:

$$\Pi = \lim_{t \to \infty} P(t)$$

L'évaluation quantitative des modèles dynamiques aura donc pour but de caractériser le vecteur P ou Π pour les états décrits dans ces représentations.

II.2.1.2 Classification des Processus Etudiés

Une propriété fondamentale pour l'évaluation quantitative des modèles de Sûreté de Fonctionnement est l'hypothèse Markovienne. Lorsqu'elle est vérifiée, elle permet de simplifier considérablement les calculs menant aux vecteurs probabilité. Nous la rappelons avec la Définition II.3.

Définition II.3 **Processus de Markov**

Soit $(X(t))_{t\in T}$ un processus stochastique à temps continu et à espace d'état discret, $(X(t))_{t\in T}$ est un *processus de Markov* si:

$$\forall (t_0, t_1, \ldots, t_n, t_{n+1}) \text{ tels que } t_0 < t_1 < \ldots < t_n < t_{n+1}$$

alors

$$P[X(t_{n+1}) \leq x_{n+1} / X(t_n) \leq x_n, \ldots, X(t_1) \leq x_1, X(t_0) \leq x_0] = P[X(t_{n+1}) \leq x_{n+1} / X(t_n) \leq x_n]$$

❑

Cette hypothèse traduit le fait que les états successifs décrivant le chemin menant dans l'état courant et le temps passé dans ces différents états par le système n'a aucune incidence sur son évolution future. Autrement dit son évolution ne dépend que de l'état présent, du temps passé dans cet état (temps local) et du temps courant (temps global). La liste suivante présente une classification des processus stochastiques par rapport à cette hypothèse fondamentale:

- Si l'évolution d'un processus de Markov est indépendante du temps local et du temps global alors ce processus est le plus simple des processus de Markov, il est appelé *Processus de Markov Homogène.*

- Si l'évolution d'un processus de Markov ne dépend pas du temps local, mais peut dépendre du temps global, alors ce processus est appelé *Processus de Markov non Homogène.*

- Enfin, si l'évolution d'un processus de Markov dépend du temps local, ce processus est appelé *Processus Semi-Markovien.* Ce processus sera *Semi-Markovien Homogène* s'il ne dépend pas du temps global et *Semi-Markovien non Homogène* dans le cas contraire.

- Tout autre processus est un processus non-Markovien.

A noter que tout processus de Markov se retrouve dans le cas homogène aux instants de changement d'état. Ces instants sont appelés points de régénération.

[Boyd 96] illustre cette classification en imaginant une grenouille sautant de nénuphar en nénuphar sur un étang. Nous nous permettons de lui emprunter cette charmante image. La

présence de la grenouille sur un nénuphar donne l'état du système et le temps passé sur un nénuphar correspond au temps de séjour dans l'état.

- *Processus Markovien.* Le prochain nénuphar atteint par la grenouille ne dépend que du nénuphar où elle est (sa localisation par rapport aux autres nénuphars, la date à laquelle la grenouille est entrée et le temps qu'elle y a déjà passé dessus) mais pas de la trajectoire qui lui a permis de l'atteindre.

- *Processus Markovien Homogène.* La durée avant le départ de la grenouille de son nénuphar est indépendante de la durée déjà passée dessus depuis qu'elle l'a atteint.

- *Processus Markoviens non Homogènes.* Idem que précédemment mais la durée de séjour peut dépendre du temps écoulé depuis le début de sa balade. S'il commence à faire nuit elle pourra quitter son nénuphar plus rapidement.

- *Processus Semi-Markovien.* Maintenant le temps restant à passer sur le nénuphar va être fonction du temps déjà passé et du temps global écoulé depuis le début pour les *Processus Semi-Markovien non Homogène.*

Pour pouvoir décrire un processus Markovien, il est important de pouvoir caractériser la probabilité de passage d'un état à un autre état. Cette probabilité s'exprime à partir du taux de transition entre deux états dont on rappelle la définition.

Définition II. 4 ***Taux de Transition et Probabilité de Transition***

Le *taux de transition* d'un état X_i à un état X_j pour un processus de Markov est défini par:

$$\lambda_{ij}(t) = \lim_{dt \to 0} \frac{1}{dt} . P\left[X(t+dt) = X_j \,/\, X(t) = X_i\right] \qquad \textbf{(II.3)}$$

La *Probabilité de Transition* de l'état X_i à l'état X_j pendant le temps dt s'exprime alors par:

$$p_{ij}(dt) = \int_0^{dt} \lambda_{ij}(u).du \qquad \textbf{(II.4)}$$

□

II.2.1.3 Processus Markoviens Homogènes

Comme on vient de le voir, les processus Markoviens homogènes sont ceux pour lesquelles les hypothèses probabilistes sont les plus éloignées de la réalité. La contrepartie de cette contrainte est que les calculs donnant accès au vecteur probabilité sont grandement simplifiés. Ceci est notamment du aux expressions résultantes des taux de transition entre états et des probabilités de passage d'un état à un autre état.

Pour un processus de Markov homogène, le temps de séjour dans un état est indépendant du temps déjà passé dans cet état. Dans ce cas, la variable aléatoire décrivant ce temps de séjour est exponentiellement distribuée (de paramètre λ). On a alors, pour deux états X_i et X_j :

$$P\left[X(t+dt) = X_j \,/\, X(t) = X_i\right] = 1 - exp(-\lambda.dt)$$

L'expression (II.3) donnant le taux de transition devient:

$$\lambda_{ij}(t) = \lim_{dt \to 0} \frac{1}{dt}.\left(1 - \exp(-\lambda.dt)\right)$$

En remplaçant $exp(-\lambda.dt)$ par son développement limité au premier ordre, on obtient:

$$\lambda_{ij}(t) = \lambda = cte$$

De même la probabilité de passage de l'état X_i à l'état X_j (pour $i \neq j$) devient:

$$p_{ij}(0,dt) = p_{ij}(t,dt) \cong \lambda_{ij}.dt$$

Vu les simplifications possibles dans l'expression des taux de transition et des probabilités de passage entre états, il semble séduisant de pouvoir appliquer l'hypothèse de processus de Markov homogène aux systèmes étudiés. Cette approximation est souvent acceptable dans le cadre des études de Disponibilité. Deux raisons à cela.

Une première justification est liée aux résultats des études statistiques donnant l'évolution du taux de défaillance d'un composant (taux de transition entre l'état de bon fonctionnement et celui de défaillance) en fonction du temps. On s'est aperçu que ce taux présente, notamment pour les composants électroniques, une courbe $\lambda(t)$ selon la Figure II.13 dite « courbe en baignoire ». Les composants utilisés dans le domaine spatial ont préalablement été vieillis, si bien que l'on peut considérer que, pendant la durée de la mission, leur taux de défaillance est constant (partie « vie utile » du composant).

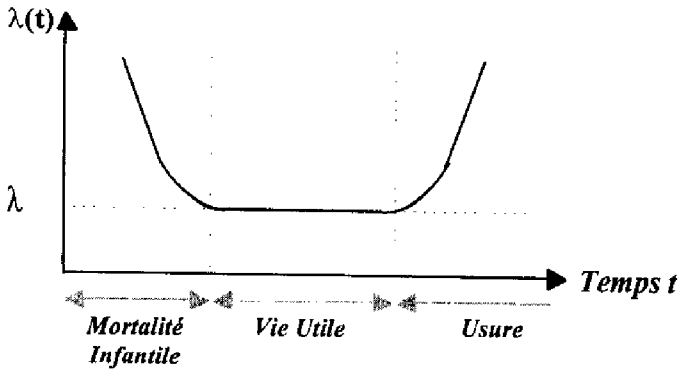


Figure II.13: « Courbe en Baignoire » du Taux de Défaillance de Composants

Une autre justification à l'utilisation des processus de Markov homogènes est liée au type de système étudié et à son stade de développement. Pour des systèmes à fonctionnement répétitif (cyclique), ne présentant pas de régime transitoire singulier et en phase très amont de développement, la modélisation par processus de Markov homogène est particulièrement bien adaptée. En effet, décrire les durées inter-état par des distributions exponentielles permet d'obtenir des résultats en valeur moyenne qui conservent le sens des variations du système par rapport aux paramètres.

II.2.1.4 Evaluation des Processus

De façon générale il existe deux moyens pour l'étude d'un processus stochastique: l'approche probabiliste qui, par une mise en équation et résolution de cette équation, permet d'avoir accès aux grandeurs exactes caractérisant le processus, ou l'approche statistique qui ne fournit que des estimations de ces grandeurs par simulation.

On vient de présenter un certain nombre de processus définis par rapport à l'hypothèse Markovienne. Cette hypothèse permet des simplifications importantes dans leur mise en équation. C'est particulièrement le cas pour les processus de Markov homogènes et on a vu que de tels processus peuvent décrire, sans induire trop d'approximations, certains systèmes dont on cherche à étudier la Disponibilité. Nous présentons donc maintenant, dans une partie intitulée « Evaluation Probabiliste », le principe de l'évaluation analytique d'un processus de Markov homogène. Nous évoquerons également certaines approches dans le cas non Markovien homogène.

L'hypothèse Markovienne est cependant très restrictive et, souvent, elle ne pourra être appliquée aux systèmes étudiés. D'autre part, même lorsqu'elle est envisageable, les approches probabilistes peuvent être déficientes (nous signalerons ces limitations). Dans ce cas, on a recours à la simulation. Cette technique est présentée dans une seconde partie appelée « Evaluation Statistique ».

II.2.2. Evaluation Probabiliste

Nous ne présentons ici que les approches probabilistes menées à partir des modèles Etats-Transitions. Il est toutefois possible d'évaluer la Disponibilité, soit par calcul direct soit à partir des modèles combinatoires, mais, contrairement aux évaluations de Fiabilité, le calcul devient rapidement très complexe et limité à des systèmes simples. Le lecteur désirant cependant avoir un aperçu de ces techniques pourra se reporter à [Pagès 80] et [Villemeur 88]. Notons, par ailleurs, que la conversion d'un modèle combinatoire à un modèle Etats-Transitions est toujours possible. C'est l'approche généralement employée par les outils informatiques basés sur les modèles combinatoires pour les évaluations de Disponibilité (se reporter par exemple à [Dugan 92]).

[Boyd 96] est un aperçu général très bien fait et très didactique sur l'utilisation des modèles Etats-Transitions pour les études quantitatives de Sûreté de Fonctionnement. Une approche plus formelle peut être trouvée dans [Doyon 89].

Les modèles Etats-Transitions sont particulièrement bien adaptés pour représenter les processus de Markov. En effet, ces derniers sont pleinement décrits à partir de ces modèles dès lors que l'on est en mesure de caractériser les probabilités de passage d'un état à un autre état. Ces probabilités, comme on l'a vu, s'expriment à partir des taux de transitions entre états (cf. expressions (II.3) et (II.4)).

Ainsi, on représente graphiquement un processus de Markov à partir d'un modèle Etats-Transitions en étiquetant chaque arc entre un état X_i et un état X_j du graphe Etats-Transitions par la valeur du taux de transition correspondant. Ces graphes sont alors appelés *Graphes de Markov*.

De même, la représentation matricielle d'un processus de Markov ayant n états est obtenue à partir des matrices Etats-Transitions en appliquant les principes suivants:

- $t_{ij}(t) = \begin{cases} \lambda_{ij}(t) & \text{si la transition de } X_i \text{ à } X_j \text{ existe} \\ 0 & \text{sinon} \end{cases}$, pour $i \neq j$
- $t_{ii}(t) = -\sum_{\substack{j=1 \\ j \neq i}}^{n} \lambda_{ij}(t)$

On appelle la matrice ainsi construite *Matrice des Taux de Transitions* ou *Générateur Infinitésimal*, on la note Λ.

II.2.2.1 Processus de Markov Homogènes

Pour de tels processus, les taux de transition entre états sont constants et s'identifient aux paramètres constants des lois exponentielles régissant les dates de franchissement entre états. On retrouve ces taux comme étiquette des arcs des graphes de Markov et comme éléments des Matrices des Taux de Transitions. Illustrons cette description en reprenant l'Exemple II.3.

Exemple II.3 (suite)

On suppose que le système précédemment décrit (système avec éléments indifférenciés) est régi par un processus de Markov homogène. Chaque événement (ou transition entre états) est donc exponentiellement distribué. La liste suivante donne les paramètres des lois exponentielles correspondantes:

- λ_{on} : paramètre de l'événement *Don* (défaillance d'un satellite actif)
- λ_{sb} : paramètre de l'événement *Dsb* (défaillance d'un satellite en attente)
- μ_{on} : paramètre de l'événement *Ron* (remplacement en mode actif d'un satellite)
- μ_{sb} : paramètre de l'événement *Rsb* (remplacement en mode attente d'un satellite)
- γ_c : paramètre de l'événement *C.* (commutation entre deux satellites)

Les représentations du processus de Markov homogène de ce système par graphe de Markov ou Matrice des Taux de Transitions sont données par la Figure II.14.

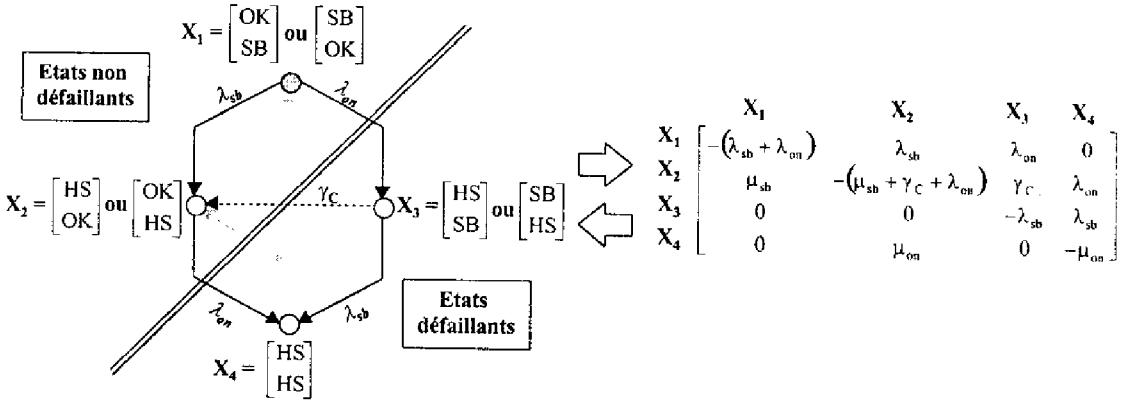


Figure II.14: Exemple de Graphe de Markov et de Matrice de Taux de Transitions

❑

La mise en équation du processus de Markov homogène revient à donner une expression du vecteur probabilité instantané (cf. Définition II.2). Etablissons donc une telle expression.

On a vu que:

$$p_{ij}(0,dt) = p_{ij}(t+dt) \cong \lambda_{ij}.dt$$

Sachant que : $\sum_{i=1}^{n} p_{ij}(dt) = 1$, on a:

$$p_{ii}(dt) = 1 - \sum_{\substack{i=1 \\ i \neq j}}^{n} p_{ij}(dt)$$

Soit $P_i(t)$, la probabilité pour que l'on soit dans l'état i au temps t. L'évolution de $P_i(t)$ pour un intervalle de temps dt s'écrit alors:

$$P_i(t+dt) = \sum_{j=1}^{n} p_{ji}(dt).P_j(t) = \sum_{\substack{j=1 \\ j \neq i}}^{n} p_{ji}(dt).P_j(t) + p_{ii}(dt).P_i(t)$$

$$P_i(t+dt) \cong \sum_{\substack{j=1 \\ j \neq i}}^{n} \lambda_{ji}.dt.P_j(t) + \left(1 - \sum_{\substack{i=1 \\ i \neq j}}^{n} \lambda_{ji}.dt\right) P_i(t)$$

$$\frac{P_i(t+dt)-P_i(t)}{dt} \cong \sum_{\substack{j=1 \\ j\neq i}}^{n} \lambda_{ji}.P_j(t) - \left(\sum_{\substack{j=1 \\ j\neq i}}^{n} \lambda_{ji}\right).P_i(t)$$

Par passage à la limite, en supposant que $P_i(t)$ est différentiable, on obtient:

$$\dot{P}_i(t) \cong \sum_{\substack{j=1 \\ j\neq i}}^{n} \lambda_{ji}.P_j(t) - \left(\sum_{\substack{j=1 \\ j\neq i}}^{n} \lambda_{ji}\right).P_i(t)$$

D'après la définition des coefficients de la Matrice des Taux de Transitions t_{ij}, on a:

$$\dot{P}_i(t) \cong \sum_{\substack{j=1 \\ j\neq i}}^{n} t_{ji}.P_j(t) + t_{ii}.P_i(t) = \sum_{j=1}^{n} t_{ji}.P_j(t)$$

D'où l'écriture matricielle de l'équation différentielle régissant l'évolution du vecteur probabilité instantanée P:

$$\dot{P}(t) = P(t) \times \Lambda \qquad \textbf{(II.5)}$$

Cette équation représente un système d'équations différentielles du premier ordre. Différentes techniques de résolution existent sachant que l'on connaît les probabilités d'états initiales $P(0)$. On peut citer :

- La résolution à partir des techniques d'Algèbre Spectrale qui s'appuie sur le calcul des valeurs propres de Λ. La solution s'exprime alors à l'aide de l'exponentielle de la matrice Λ:

$$P(t) = P(0).exp(\Lambda.t)$$

- La résolution à l'aide de la transformée de Laplace de l'équation (II.5). On a alors, sachant que: $L[P(t)] = P(s)$

$$P(t) = L^{-1}[P(s)] = L^{-1}[P(0).\{sI - \Lambda\}^{-1}]$$

Si l'équation (II.5) est très puissante car elle donne accès à la probabilité au cours du temps de chaque état du processus, sa résolution peut s'avérer très lourde à mettre en œuvre et pas forcément utile dans le cas de systèmes répétitifs. En effet, pour ces derniers, l'étude du régime permanent est beaucoup plus simple à appréhender. Mais se pose le problème, d'une part, de l'existence d'un régime permanent et, d'autre part, de l'unicité de ce dernier.
L'existence du régime permanent est assurée pour les processus de Markov à nombre d'états finis par un premier résultat affirmant qu'il existe toujours, pour de tels processus, au moins une distribution stationnaire.

L'unicité du régime permanent peut être établie à partir de l'étude du comportement logique du graphe de Markov. En effet, les états d'un graphe de Markov peuvent se classer selon deux familles:

- Les *états transitoires* dont on peut sortir mais vers lesquels il est impossible de retourner et
- Les *états ergodiques* dont il n'est plus possible de sortir une fois atteints.

L'étude du régime stationnaire du processus dépendra donc du nombre de classes d'états ergodiques. Remarquons que ces classes correspondent au nombre de composantes fortement connexes ajouté au nombre d'états puits du graphe de Markov. Un résultat important permet alors d'affirmer qu'un processus de Markov à nombre d'états fini admet une unique distribution stationnaire s'il ne possède qu'une seule classe d'états ergodiques. Le processus est alors appelé *processus de Markov Ergodique.*

Dans la Figure II.14, le graphe de Markov est fini, donc il existe au moins une distribution stationnaire. Cette dernière est unique car le graphe ne possède qu'une seule composante fortement connexe qui recouvre tous les états. Dans ce cas particulier où la classe unique des états ergodiques recouvre tous les états, on dit que le processus de Markov est *irréductible.*

Le calcul de la distribution stationnaire pour un processus de Markov ergodique se fait à partir de l'équation (II.5) sachant que l'on a :

$\Pi = [\Pi_1, .., \Pi_n]$: vecteur probabilité stationnaire unique et $\dot{P}(t) \to 0$ *pour* $t \to \infty$

L'équation (II.5) devient alors:

$$\Pi \times \Lambda = 0 \qquad avec \quad \sum_{i=1}^{n} \Pi_i = \qquad \textbf{(II.6)}$$

Exemple II.4

Supposons un satellite initialement en bon fonctionnement et pouvant être remplacé en cas de défaillance, on admet qu'il y a toujours un satellite de secours disponible. On note λ et μ respectivement le taux de défaillance et le taux de remplacement. L'hypothèse de processus Markovien homogène impose λ et μ constants. La représentation par modèle Etats-Transitions proposée par la Figure II.1 est alors triviale. On note X_1 l'état pour lequel il y a un satellite actif et X_2 l'état pour lequel il n'y en a pas.

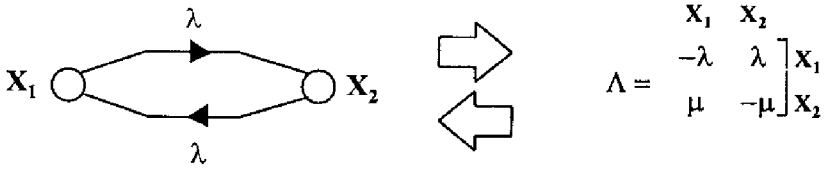


Figure II.1: Graphe de Markov et Matrice des Taux de Transitions

La résolution de l'équation (II.5) nous donne alors les probabilités respectives de X_1 et X_2 au cours du temps. En supposant qu'on est initialement dans l'état X_1, on obtient:

$$\begin{cases} P_{X_1}(t) = \dfrac{\mu}{\lambda+\mu} + \dfrac{\lambda}{\lambda+\mu}.exp\left[-(\lambda+\mu)t\right] \\ P_{X_2}(t) = \dfrac{\lambda}{\lambda+\mu} + \dfrac{\mu}{\lambda+\mu}.exp\left[-(\lambda+\mu)t\right] \end{cases}$$

L'état X_1 est l'état de disponibilité, donc la disponibilité du système en fonction du temps vaut:

$$A(t) = P_{X_1}(t) = \frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu}.exp\left[-(\lambda+\mu)t\right]$$

La disponibilité stationnaire peut être calculée car ce processus de Markov est irréductible donc ergodique. Un premier moyen est d'utiliser l'équation (II.6), mais, disposant déjà de $A(t)$, on l'obtient directement par:

$$A(\infty) = \lim_{t\to\infty} A(t) = \lim_{t\to\infty} P_{X_1}(t) = \Pi_{X1} = \frac{\mu}{\lambda+\mu}$$

❑

Deux problèmes majeurs se posent pour la résolution numérique des équations (II.5) et (II.6). Le premier est lié à la taille mémoire nécessaire: elle dépend du nombre d'états qui, comme on l'a vu, peut devenir rapidement très important. De nombreuses recherches portent sur cette problématique et visent à simplifier les équations (II.5) et (II.6) soit en exploitant les parties « creuses » des matrices A, soit en utilisant des techniques de regroupements d'états (agrégation Markovienne [Kemeny 76]) ou de troncature d'états.

Le deuxième problème concerne la précision numérique des résultats. Les taux de transitions mis en jeu dans la matrice A peuvent différer de plusieurs ordres de grandeur, c'est par exemple le cas entre le taux caractérisant un événement de défaillance (qui peut se produire, en valeur moyenne, au bout de plusieurs mois voire années) et celui représentant un événement de commutation sur un élément en attente (qui peut intervenir, en valeur moyenne, après quelques heures seulement). De tels processus de Markov sont appelés « Stiff Markov Chains » (bien qu'en fait il s'agisse de processus de Markov et non de chaînes de Markov). La résolution numérique de tout système contraignant à faire des approximations (codage des réels) elles deviennent critiques dans le cas des « Stiff Markov Chains ». Ce problème est traité soit par des techniques numériques [Reibman 88], soit par décomposition hiérarchique des modèles en fonction de l'échelle de temps (« Time Scale Decomposition ») [Courtois 77], il s'agit alors de techniques d'approximation.

Même s'il existe des solutions, ces deux problèmes constituent les principales limitations au traitement quantitatif des modèles Etats-Transitions décrivant des processus de Markov homogènes.

II.2.2.2 Autres Processus

Comme on a pu l'évoquer (cf. II.2.1.3.) l'hypothèse de processus de Markov homogènes pour le traitement quantitatif des modèles de Sûreté de Fonctionnement est souvent acceptable. C'est le cas pour les études de Fiabilité au niveau sous-système où les taux de défaillance des composants peuvent être supposés constants, c'est également le cas pour les systèmes répétitifs en phase très amont de développement. Cependant, certaines caractéristiques des systèmes étudiés peuvent nous obliger à rejeter cette hypothèse qui ne permet plus de décrire des phénomènes incontournables pour l'évaluation de grandeurs comme la Disponibilité. Parmi ces caractéristiques on peut citer:

- La défaillance de certains sous-systèmes dont le taux de défaillance équivalent n'est plus constant,
- Les processus d'initialisation et de maintenance dont la durée dépend du temps déjà écoulé depuis leur déclenchement,
- La fin de vie « naturelle » d'éléments (ex: un satellite ayant consommé tout son ergol) qui dépend du temps écoulé depuis leur mise en service, ou encore
- Les phases de mission dont la durée dépend d'un temps global initié au début de la mission.

Dans certains cas on aura besoin de mémoriser le temps passé dans un état ou le temps écoulé depuis le début de l'évolution du système. L'hypothèse Markovienne reste alors vérifiée mais les processus ne sont plus homogènes (non homogènes, semi-Markoviens cf. §II.2.1.2). Des approches analytiques restent toutefois envisageables mais pour des systèmes de petites tailles et souvent en régime permanent. On peut citer:

- *La méthode des Etats Fictifs*. Elle consiste à remplacer toute transition à taux non constant entre deux états par une série d'états fictifs reliés entre eux par des transitions, elles, à taux constants. Si la date de transition à taux non constant à reproduire a pour valeur moyenne $1/\lambda$, pour n états fictifs créés, les transitions entre ces états auront pour taux constant $n \times \lambda$. La valeur moyenne reste conservée et l'écart type (en $1/\lambda$ pour les distributions exponentielles) devient en $1/(\lambda \times \sqrt{n})$. Cette méthode permet donc de centrer les dates de transition autour de leur valeur moyenne mais au prix d'une augmentation importante du nombre d'états car l'écart type diminue en $1/\sqrt{n}$. Elle n'est donc généralement pas applicable pour des systèmes de « taille réelle ».
- Les Techniques de Forçage d'Etats. Elles sont utilisées pour représenter les phases de mission où, pour chaque phase, le système est supposé Markovien homogène.
- La Méthode de la Chaîne Immergée. Cette méthode est notamment utilisée pour les processus semi-Markoviens homogènes et se base sur les instants où l'état du processus résume toute son évolution passée (appelés points de régénération), c'est à dire est comparable à un processus Markovien homogène. L'extraction de ces instants particuliers permet de générer une chaîne de Markov (l'espace des temps est alors discret, c'est l'ensemble des points de régénération) dont l'étude peut fournir certaines caractéristiques du processus.

Ces méthodes supposent que l'hypothèse Markovienne est vérifiée, ou en partie, mais pour les systèmes dont on vient d'énoncer certaines caractéristiques embarrassantes, ce n'est rapidement plus le cas. Une des principales raisons est liée au parallélisme et aux synchronisations inhérentes à l'évolution des éléments de tels systèmes. Illustrons ceci sur un exemple simple.

Exemple II.5

Supposons un satellite actif pouvant défaillir avec un taux de défaillance constant λ. S'il défaille, il est remplacé par un satellite de secours, au bout d'un durée décrite par une variable aléatoire « à mémoire ». On suppose, en outre, que le satellite de secours ne peut pas défaillir tant qu'il n'est pas actif. La Figure II.2 décrit une partie du modèle Etats-Transitions correspondant. On note « *S.* » le satellite nominal et « *Sec.* » le satellite de secours.

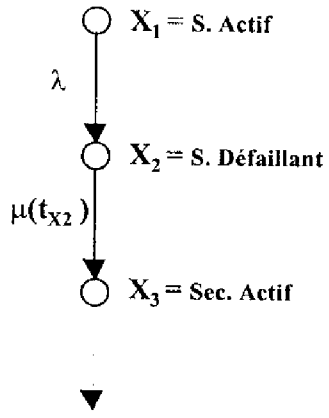


Figure II.2: Modèle Etats-Transitions du Premier Système

L'hypothèse Markovienne reste vérifiée car une fois dans l'état X_2 le taux de transition μ vers X_3 ne dépendra que du temps passé dans X_2 : $\mu(t_{X2})$.C'est alors un processus semi-Markovien homogène.

Supposons maintenant le même système mais avec un satellite actif supplémentaire. La Figure II.3 décrit l'évolution du système correspondant au remplacement du premier satellite en cas de défaillance. On note « *S1.* » et « *S2.* » les deux satellites nominaux.

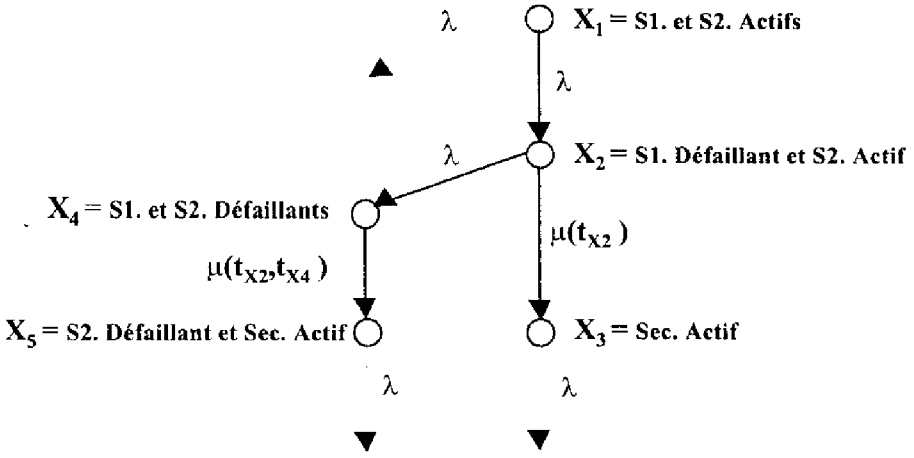


Figure II.3: Modèle Etats-Transitions du Second Système

Lorsque le système est dans l'état X_2, soit rien ne se passe pendant qu'il est remplacé par le satellite de secours. On est alors dans le même cas que précédemment et le taux de transition vers X_3 ne dépend que de t_{X2}. Soit avant d'être remplacé, le second satellite défaille. Le système est alors dans un nouvel état: X_4. La transition vers l'état pour lequel le satellite de secours est activé dépendra alors du temps déjà passé dans l'état X_2 ainsi que du temps passé dans le nouvel état X_4 , soit $\mu = \mu(t_{X2}, t_{X4})$. En effet, la manœuvre de remplacement n'est pas reprise à zéro parce que le second satellite a défailli. Dans ce cas, le processus n'est plus un processus de Markov.

❑

Les méthodes analytiques, pour de tels systèmes, se trouvent alors rapidement mises en échec. On peut toutefois citer la *Méthode des Variables Complémentaires* dont l'idée sous-jacente est d'associer aux états des variables supplémentaires décrivant le temps écoulé depuis l'arrivée dans certains états. Les taux de transitions sont alors fonction de ces variables. Dans l'Exemple II.5, on associerait aux états X_2 et X_4 la variable θ mesurant le temps écoulé depuis le début de l'activation du satellite de secours c'est à dire depuis l'entrée dans l'état X_2. Mais la résolution de tels modèles est complexe et se trouve limitée par le nombre de variables supplémentaires introduites.

II.2.3. Evaluation Statistique

Les techniques de simulation sont un procédé déjà ancien pour l'évaluation et le dimensionnement de systèmes [Agard 68]. Et de nos jours, il n'est pas un domaine d'ingénierie qui n'y ait recours. La simulation est justifiée lorsque les approches analytiques sont mises en échec. C'est le cas quand on ne peut plus les mettre en œuvre (mise en équation trop complexe, problèmes liés à leur informatisation), mais également lorsque les hypothèses sous-jacentes (comme l'hypothèse Markovienne) sont trop restrictives pour décrire le comportement de systèmes complexes. Et pour les études de Disponibilité, comme on a pu le souligner dans ce qui précède, ces limitations sont rapidement atteintes. Mais il est important d'insister ici sur le fait que le recours à la simulation ne doit venir qu'en dernier lieu car, en

tant qu'approche statistique, elle n'offre que des estimations des grandeurs étudiées et souvent avec des temps de simulation beaucoup moins performants que les temps de calcul.

[Law 91] est une étude approfondie les différentes techniques de simulation utilisées pour le dimensionnement de systèmes. Nous restreignons ici notre propos aux évaluations de grandeurs de Sûreté de Fonctionnement en présentant les principes sous-jacents.

II.2.3.1 Paramètres de Simulation

La simulation, de façon très générale, se base sur l'observation statistique de l'évolution de modèles logiques qui décrivent le comportement de systèmes complexes. Dans le cas de la simulation de processus stochastiques, on obtient alors des estimations sur les grandeurs caractérisant ces processus. La précisions sur les estimations réalisées dépend de deux paramètres fondamentaux: le nombre d'histoire N, la durée de chaque histoire T. Précisons donc leur impact sur la qualité des résultats de simulation.

L'évaluation analytique d'un processus stochastique à espace d'état discret et fini rend compte de toutes ses évolutions possibles au cours du temps. A savoir:

- Les différents changements d'états, c'est-à-dire tous les chemins du graphe d'états,
- Les différentes dates de transition entre états.

Ceci est résumé dans l'expression du vecteur probabilité instantané $P(t)$ qui donne la valeur exacte de ces probabilités d'états au cours du temps (cf. II.2).

La simulation ne peut fournir quant à elle qu'une estimation de ce vecteur. En effet, elle rend compte de la valeur exacte de l'état du modèle au cours du temps mais pour une seule évolution du processus appelée *histoire* ou *trajectoire*. L'estimation de la valeur de $P(t)$ nécessite alors un nombre important d'histoires. Ceci afin de:

- Parcourir tous les chemins possibles du graphe d'états (possible car l'espace d'état est supposé fini),
- Générer des dates de changement d'états significatives (les générer toutes est impossible car l'espace des temps est continu).

Le nombre d'histoires N est donc un paramètre important d'une simulation. Pour les études en régime transitoire, la précision des résultats de l'estimateur par rapport à la grandeur qu'il estime est conditionnelle à N.

Pour les études en régime stationnaire ce paramètre est moins prépondérant. En effet, si le processus stochastique est ergodique, on sait que toute trajectoire permet d'atteindre au bout d'un certain temps l'unique régime stationnaire c'est à dire la classe des états ergodiques. Une histoire peut donc suffire. Dans ce cas, c'est la durée de simulation T qui conditionne la précision des résultats, car, une fois les états ergodiques atteints, plus cette durée est importante plus on peut visiter un nombre significatif de fois ces états et ainsi obtenir une estimation de leur probabilité stationnaire.

Les techniques de simulation peuvent donc différer selon qu'on s'intéresse au régime transitoire ou stationnaire. La fin d'une simulation est dans tous les cas liée à la précision souhaitée sur les estimateurs. Cette précision dépendra de N et/ou de T.

II.2.3.2 Génération de Nombres Aléatoires

Pour pouvoir décrire une évolution d'un processus stochastique, il faut être en mesure de simuler les variables aléatoires correspondant aux dates de changements d'état. La technique généralement employée consiste à utiliser un générateur aléatoire donnant une valeur uniformément répartie sur l'intervalle *[0,1]*. Notons qu'un tel générateur doit satisfaire les propriétés suivantes afin d'être utilisable :

- Il doit avoir une période la plus importante possible,
- Les nombres générés ne doivent pas être corrélés entre eux,
- Les séquences de nombres générés doivent être reproductibles,
- Il ne doit pas nécessiter une importante place mémoire et doit être suffisamment rapide.

Le nombre aléatoire ainsi généré est assimilé à la probabilité de la variable aléatoire. Pour connaître cette variable en fonction de sa loi, on applique au nombre généré l'inverse de la fonction de répartition décrivant cette variable aléatoire.

Exemple II.6

Supposons que l'on veuille simuler une date aléatoire θ exponentiellement répartie de paramètre λ. On génère donc un nombre entre *[0,1]*, soit π le résultat. La fonction de répartition θ s'écrit:

$$F(\theta) = 1 - exp(-\lambda.\theta)$$

π correspond à la valeur de $F(\theta)$ pour une certaine valeur de θ que l'on obtient par:

$$\theta = F^{-1}(\pi) = -\frac{1}{\lambda}.ln(1-\pi)$$

❑

II.2.3.3 Mise en Œuvre

La mise en œuvre de la simulation se fait à partir de modèles décrivant la logique d'évolution des systèmes à évaluer. Pour les études de Disponibilité, ces modèles peuvent être ceux déjà présentés au §II.1 ou des programmes informatiques spécialement écrits pour représenter un système donné. L'évolution de tels modèles n'a lieue qu'aux instants aléatoires générés (selon le principe décrit ci-avant) pour représenter les changements d'état. La technique généralement employée est celle de la simulation dirigée par événements par opposition à la simulation dirigée par horloge. Illustrons ce principe.

Au début de la simulation, un certain nombre de dates aléatoires, appelées événements, sont générées (selon la méthode déjà décrite). Ces événements correspondent aux instants possibles de départ de l'état initial du modèle. Ils sont empilés dans un échéancier du plus proche de la date courante au plus éloigné. Le temps est alors avancé jusqu'à l'événement du haut de la pile, le changement d'état a lieu et, une fois dans ce nouvel état, la pertinence des événements encore dans l'échéancier est examinée (certains événements peuvent être retirés). Les événements correspondant à l'état courant sont générés et insérés dans l'échéancier. La temps est à nouveau avancé jusqu'à l'événement du haut de pile et le procédé déjà décrit se reproduit. L'utilisation d'une horloge n'est donc pas nécessaire.

II.2.3.4 Estimateurs

Nous définissons ici un certain nombre d'estimateurs que peuvent proposer les approches de simulation et en relation directe avec les grandeurs de Sûreté de Fonctionnement. On notera E un ensemble d'états inclus dans l'espace d'état du processus simulé ($E \subset X$) et N le nombre d'histoires réalisées.

Un premier estimateur, incontournable pour l'évaluation de la Disponibilité, est l'estimateur du vecteur probabilité instantané pour l'ensemble E défini par :

$$\psi_E(\theta) = P\left[X(\theta) \in E \ / \ X(0)\right]$$

Définition II. 5 ***Estimateur de $\psi_E(\theta)$***

L'estimateur de $\psi_E(\theta)$, noté $\hat{\psi}_E(\theta)$, est défini par:

$$\hat{\psi}_E(\theta) = \frac{1}{N} \cdot \sum_{k=1}^{N} U_k, \quad \text{avec} \quad U_k = \begin{cases} 1 \text{ si } X^k(\theta) \in E \\ 0 \text{ sinon} \end{cases} \qquad \textbf{(II.7)}$$

❑

Les variables U_k sont les variables booléennes mesurant pour chaque histoire k, à la date θ, l 'appartenance de l'état courant $X^k(\theta)$ à l'ensemble E. L'estimateur $\hat{\psi}_E(\theta)$ est donc défini comme la valeur moyenne de ces variables pour les N histoires réalisées.

Si on considère que E est l'ensemble des états pour lesquels le système est disponible, alors l'estimation de la disponibilité instantanée sera directement donnée par :

$$\hat{A}(\theta) = \hat{\psi}_E(\theta)$$

Un deuxième estimateur permet d'avoir accès, cette fois, à une estimation de la Fiabilité. C'est celui qui évalue la probabilité de rester dans un ensemble d'états E depuis le début de la simulation définie par :

$$\zeta_E(\theta) = P\left[\forall \alpha \in [0,\theta],\ X(\alpha) \in E \ / \ X(0)\right]$$

Définition II. 6 ***Estimateur de $\zeta_E(\theta)$***

L'estimateur de $\zeta_E(\theta)$, noté $\hat{\zeta}_E(\theta)$, est défini par:

$$\hat{\zeta}_E(\theta) = \frac{1}{N} \cdot \sum_{k=1}^{N} W_k, \quad avec \quad W_k = \begin{cases} 1 \ si \ \forall \alpha \in [0,\theta] \ X^k(\alpha) \in E \\ 0 \ sinon \end{cases} \qquad \textbf{(II.8)}$$

❑

Les variables W_k sont les variables booléennes mesurant pour chaque histoire k, sur l'intervalle $[0,\theta]$, l 'appartenance ou non de l'état courant X^k à l'ensemble E. L'estimateur $\hat{\zeta}_E(\theta)$ est donc défini comme la valeur moyenne de ces variables pour les N histoires réalisées.

De même que précédemment, si l'on considère E comme l'ensemble des états de Fiabilité, alors l'estimation de la Fiabilité est directement donnée par:

$$\hat{R}(\theta) = \hat{\zeta}_E(\theta)$$

Certaines variables annexes peuvent également être estimées. On peut citer:

- Le nombre de passage dans l'ensemble des états E jusqu'à la date θ. Il s'agit alors d'une variable entière.
- Le temps de séjour cumulé dans l'ensemble E jusqu'à la date θ. C'est dans ce cas une variable continue.

Ces variables sont estimées selon la même technique. Soit x_k l'une d'entre elles, l'estimation de leur valeur moyenne et de leur écart type est donnée par :

$$\hat{m}(\theta,N) = \frac{1}{N} \cdot \sum_{k=1}^{N} x_k(\theta) \qquad et \qquad \hat{\sigma}(\theta,N) = \frac{1}{N-1} \cdot \sum_{k=1}^{N} \left[x_k^2(\theta) - \hat{m}^2(\theta,N)\right]$$

avec $x_k(\theta)$ valeur de la variable pour l'histoire k et à la date θ.

Enfin, pour les simulations dont le critère d'arrêt de chaque trajectoire est l'occurrence d'un événement particulier, on peut estimer la variable aléatoire correspondant à la date de cet événement. Il s'agit alors d'une variable temporelle dont l'estimation de la valeur moyenne et de l'écart type est donnée par :

$$\hat{m}(N) = \frac{1}{N} \cdot \sum_{k=1}^{N} \theta_k \qquad et \qquad \hat{\sigma}(N) = \frac{1}{N-1} \cdot \sum_{k=1}^{N} \left[\theta_k^2 - \hat{m}(N)\right]$$

avec θ_k *valeur de la variable temporelle pour l'histoire k.*

Si θ correspond à la date de la première défaillance du système on peut ainsi avoir accès à une estimation du MTTF (cf. I.2.2).

Pour tous ces estimateurs on doit connaître la précision avec laquelle ils sont évalués. Pour cela on utilise généralement le calcul de l'intervalle de confiance qui caractérise la précision de la mesure pour un estimateur donné. Nous rappelons la définition d'un tel intervalle dans le cas simplifié où ce dernier est symétrique.

Définition II.7 ***Intervalle de Confiance***

Soit Z un estimateur d'une quantité z fournissant une estimation $\hat{z}$. Soit α un réel, $0 < \alpha < 1$. On appelle *Intervalle de Confiance* au niveau $1 - \alpha$ l'intervalle :

$$\left[\hat{z} - \hat{\Delta z}, \hat{z} + \hat{\Delta z}\right] \text{ où } \hat{\Delta z} \text{ est défini par } P\left[|\hat{z} - z| > \hat{\Delta z}\right] < \alpha$$

❑

Généralement α vaut entre *1* et *10%*. Le calcul de ces intervalles n'est pas toujours trivial car il dépend de la nature de la quantité évaluée et du procédé de simulation. Certains résultats peuvent être trouvés dans [Leroudier 80]. Nous reviendrons sur leur expression dans le cadre particulier de la simulation des réseaux de Petri (cf. Chapitre IV).

II.2.3.5 Conclusion

Si la simulation présente un certain nombre d'inconvénients (grandeurs uniquement estimées, temps de simulation pour une précision requise), elle a avant tout le mérite de proposer des solutions à l'étude de systèmes de taille et de complexité « réelle ». Elle offre, de plus, une large palette de résultats grâce aux différents estimateurs qui peuvent être mesurés.

Un point toutefois, qu'il est important de ne pas négliger, concerne la validité des modèles que l'on simule. En effet ces modèles doivent permettre de représenter des comportements complexes pour lesquels les méthodes analytiques sont mise en échec, ils sont par conséquent complexes eux-mêmes à concevoir et à vérifier. Si on se repose sur les modèles de Sûreté de Fonctionnement (cf. §II.1), on est rapidement confronté aux limites déjà évoquées: pouvoir d'expression réduit pour les modèles combinatoires et explosion du nombre d'états pour les modèles Etats-Transitions. Si on décrit directement le comportement d'un système par un programme informatique, des erreurs peuvent être facilement introduites par le codeur et le programme obtenu sera un piètre support de communication avec l'équipe de conception du système. Ce programme peut alors décrire un comportement différent de celui espéré pour le système à concevoir sans que les membres de l'équipe de conception du système puissent s'en rendre compte.

II.3. Synthèse

Ce chapitre nous a permis de faire un tour d'horizon des méthodes « classiques » d'Etude de Disponibilité Opérationnelle. Nous avons pu évoquer les principes de ces méthodes aussi bien pour la modélisation que pour l'évaluation. Ceci nous a permis de dégager un certain nombre de caractéristiques. Etablissons donc un bref bilan.

La représentation de la logique d'évolution des systèmes repose sur deux types de modèles: les modèles combinatoires et les modèles Etats-Transitions.
Les modèles combinatoires sont simples à concevoir et à comprendre. Historiquement définis pour les études de Fiabilité, ils sont particulièrement bien adaptés pour représenter les dépendances entre les éléments d'un système. Grâce à certaines extensions, pour lesquelles la logique booléenne sous-jacente n'est plus respectée, on est en mesure de décrire des interactions encore plus finement. Cependant la notion d'état n'est pas explicitement considérée ce qui limite singulièrement leur pouvoir de modélisation. D'autres phénomènes que les défaillances (ou plus généralement les événements redoutés) sont très difficiles à représenter. Ceci est particulièrement limitatif pour les études de Disponibilité pour lesquelles les événements liés aux actions d'initialisation et de maintenance sont incontournables. On rappelle que la Disponibilité est pertinente à étudier dés lors qu'un système peut revenir d'un état défaillant dans un état de bon fonctionnement (cf. §I.3.2).
Les modèles Etats-Transitions, étant basés sur l'énumération des états du système à étudier, permettent de lever cette limite de pouvoir d'expression des modèles combinatoires. D'autre part, de nombreux résultats de la théorie des graphes offrent des possibilités de vérification formelle de certains comportements logiques. Mais pour les systèmes de grande taille, l'explosion combinatoire du nombre d'états peut être critique et l'énumération difficile (voire impossible). La conception des modèles est alors périlleuse et les modèles obtenus sont rapidement illisibles entravant de ce fait la communication entre les responsable du développement du système.

Par ailleurs, nous avons présenté les principales approches d'évaluation quantitative de la Disponibilité. Les approches probabilistes pertinentes reposent sur l'étude des modèles Etats-Transitions pour lesquels la notion de taux de transition a été définie. Ces modèles sont particulièrement bien adaptés pour représenter les processus Markoviens. La résolution analytique est performante pour les processus de Markov homogènes, bien qu'elle puisse être limitée, une fois encore, par le nombre d'états du système.
Pour les autres processus, certaines approches probabilistes ont été évoquées mais elles font intervenir des hypothèses par trop contraignantes et ne sont généralement pas applicables pour les systèmes de taille importante.
L'approche statistique a alors été présentée. Nous en avons introduit les principes en soulignant l'importance de disposer d'un modèle simulable facile à concevoir, à comprendre, à vérifier et ayant un pouvoir de modélisation important étant donnée la complexité des phénomènes à représenter qui justifient le recours à la simulation. Les modèles classiques de Sûreté de Fonctionnement aussi bien que les programmes informatiques directement écrits ne satisfont généralement pas ces contraintes.

Les méthodes « classiques » d'étude de Disponibilité présentent donc un certain nombre de limitations aussi bien du point de vue de la modélisation des systèmes que de leur évaluation quantitative. Ces limitations sont d'autant plus critiques que les systèmes étudiés

sont de grande taille, avec des comportements dynamiques complexes et pour lesquels l'hypothèse Markovienne ne peut s'appliquer. Or, les systèmes spatiaux qui nécessitent des études de Disponibilité, tombent rapidement dans cette catégorie.
Un premier exemple peut être l'étude de la Disponibilité d'une station de poursuite de satellites géographiquement isolée (comme la station du CNES des îles Kerguelen). Si on a la possibilité de décrire l'évolution de ce système par un processus de Markov homogène en régime permanent, le nombre d'éléments peut être considérable et les phénomènes liés à la maintenance complexes (commandes de matériel à la métropole, utilisation des stocks, de ressources humaines,...). Une approche probabiliste est envisageable (moyennant des solveurs Markoviens performants) mais le modèle Etats-Transitions est alors quasiment impossible à construire « à la main ».
On peut également reprendre l'exemple d'un segment spatial de constellation de satellites. Un tel système peut présenter un grand nombre de satellites avec des interactions fortes (redondance en orbite) et des processus de déploiement et de maintenance complexes (échéancier de déploiement, politique de remplacement suite à panne ou fin de vie). De plus, on souhaite essentiellement l'étudier en régime transitoire: il est important, en effet, d'avoir la connaissance des phases de déploiement et de renouvellement. Enfin, les hypothèses Markoviennes ou semi-Markoviennes ne peuvent s'appliquer car on doit pouvoir représenter des phénomènes comme la durée de vie des satellites ou encore les durées des actions correctrices (on retombe dans le type de problème décrit dans l'Exemple II.). L'unique approche envisageable est dans ce cas la simulation et on est alors confronté au problème de conception des modèles simulables.

DEUXIEME PARTIE

RESEAUX DE PETRI ET ETUDE DE DISPONIBILITE OPERATIONNELLE

La première partie de ce mémoire nous a permis de présenter le cadre particulier des études de Disponibilité Opérationnelle des Systèmes Spatiaux ainsi que les méthodes traditionnellement employées pour mener ces études. Nous avons pu ainsi mettre en évidence un certain nombre de limitations qui incitent à se tourner vers de nouvelles approches.

Cette deuxième partie est dédiée aux réseaux de Petri. Ce sera pour nous l'occasion de souligner les éléments de réponse que propose l'utilisation de ce modèle face aux limitations évoquées. Notamment :

- La maîtrise de l'explosion combinatoire du nombre d'états,
- Le pouvoir de modélisation,
- Les possibilités de vérification formelles,
- L'intelligibilité des modèles,
- La compatibilité avec les approches probabilistes présentées,
- La mise en oeuvre efficace de leur simulation.

Compte tenu de la très récente introduction des réseaux de Petri dans les activités du CNES et d'ALCATEL ESPACE où se sont déroulés ces travaux, il nous a semblé important de rappeler quelques éléments fondamentaux de cette théorie.

Pour guider notre présentation et illustrer les différents concepts, nous nous appuyons au cours des Chapitre III et IV sur l'exemple incontournable dans le cadre des études de Disponibilité : la modélisation et l'évaluation de systèmes constitués d'éléments en redondance [Ereau 94] [Ereau 97]. Par ailleurs cet exemple nous sera très utile dans le chapitre V consacré à l'étude d'un cas pratique de taille significative : la Disponibilité Opérationnelle d'une constellation de satellites.

Chapitre III

Les Réseaux de Petri

Les réseaux de Petri ont été introduits par Carl Adam Petri dans sa dissertation de thèse en 1962 (une traduction anglaise est paru en 1966 [Petri 66]). La capacité de ce modèle à décrire l'évolution des systèmes distribués a progressivement séduit la communauté internationale des chercheurs de ce domaine. Depuis les premières ébauches par Carl Adam Petri, les réseaux du même nom ont fait l'objet d'une recherche mondiale très intense si bien qu'aujourd'hui ils constituent une théorie très riche appliquée dans de nombreux domaines. Parmi eux on peut citer l'analyse des protocoles de communication, la commande des ateliers de fabrication, la conception de logiciels temps réels, la validation des interfaces homme-machine ou encore l'évaluation des performances des systèmes discrets.

Les études de Disponibilité tombant dans cette dernière catégorie, c'est bien sûr pour ce domaine d'application que porte notre intérêt pour les réseaux de Petri. Ce chapitre se centre exclusivement sur les possibilités de modélisation et de vérification offertes par ce modèle. L'évaluation quantitative des modèles fera l'objet du chapitre suivant.

La première partie du présent chapitre est consacrée à la définition des réseaux de Petri. Nous présentons ensuite les principes d'évolution du modèle. La troisième partie décrit un certain nombre de propriétés qualitatives ainsi que les moyens de les vérifier sur un réseau. Ces propriétés sont importantes car elles apportent des éléments de validation formelle qui permettent de mettre à jour des erreurs de modélisation ou de justifier la confiance que l'on porte dans ce modèle pour représenter un système donné. Enfin, nous introduisons une extension aux réseaux de Petri définis qui permet d'augmenter leur pouvoir de modélisation pour décrire plus finement un système.

Le lecteur désirant avoir une présentation plus approfondie et plus complète pourra se reporter aux ouvrages suivants: [Valette 92], [David 92], [Brams 83], [Murata 89], [Peterson 81].

II.1. Définition

II.1.1. Définition Formelle et Représentations d'un Réseau de Petri

Définition III.1 ***Réseau de Petri et Réseau de Petri Marqué***

Un *Réseau de Petri* est un quadruplé:

$$R = \langle P,\ T,\ Pre,\ Post \rangle$$

où:

- P est un ensemble fini de places de cardinal $|P|$
- T est un ensemble fini de transitions de cardinal $|T|$
- $Pre: P \times T \rightarrow N$ est l'application incidence avant (places précédentes)
- $Pre: P \times T \rightarrow N$ est l'application incidence arrière (places suivantes)

On note: $C = Post - Pre$

Un *Réseau de Petri marqué* est un couple:

$$N = (R;\ M)$$

où:

- R est un réseau de Petri
- M est le marquage du réseau, c'est à dire l'application

$$M: P \rightarrow N$$

❑

Le marquage d'un réseau donne, pour chaque place $p \in P$, un entier $M(p)$ indiquant le nombre de marques associées à p. Les marques sont également appelées jetons.

Lorsque *Pre* et *Post* sont des applications de $P \times T \rightarrow \{0,1\}$ on dit que le réseau est un *réseau de Petri Ordinaire*. Dans le cas contraire il s'agit de *réseau de Petri Généralisé*.

Notations

Soit une transition t, on note t° (resp. $^\circ t$), l'ensemble des places avales (resp. amonts) de la transition t. C'est à dire:

$$t^\circ = \{p \in P,\ Post(p,t) \neq 0\} \quad \text{et} \quad {}^\circ t = \{p \in P,\ Pre(p,t) \neq 0\}$$

De même, soit une place p, on note p° (resp. $^\circ p$), l'ensemble des transitions avales (resp. amonts) de la place p. Soit:

$$p^\circ = \{t \in T,\ Pre(p,t) \neq 0\} \text{ et } {}^\circ p = \{t \in T,\ Post(p,t) \neq 0\}$$

❑

On peut associer à un réseau de Petri un graphe biparti et fini dont les deux types de noeuds sont les places et les transitions représentées respectivement par des cercles et des rectangles. Un arc relie une transition t à une place p (resp. une place p à une transition t) si et seulement si $Pre(p,t) \neq 0$ (resp. $Post(p,t) \neq 0$). Ces arcs sont étiquetés par la valeur de $Pre(p,t)$ (ou de $Post(p,t)$) lorsque celle ci est supérieure à 1. Le marquage du réseau est donné par une répartition de jetons dans les cercles du graphe.

On peut également décrire un réseau de Petri à partir d'une représentation algébrique. *Pre*, *Post* et C sont alors des matrices de dimension $|P| \times |T|$ et le marquage M est un vecteur de dimension $|P|$.

La matrice C est appelée *matrice d'incidence*. Elle décrit complètement la structure du réseau si le graphe associé ne comprend aucune boucle élémentaire. Un tel réseau est appelé un *réseau de Petri pur*, il satisfait la condition:

$$\forall p \in P \text{ et } t \in T : Pre(p,t).Post(p,t) = 0$$

II.1.2. Exemple

La Figure III.1 illustre la représentation graphique d'un exemple de réseau de Petri marqué $N = \langle R, M \rangle$. Ce graphe ne présente pas d'arc valué. C'est à dire que l'on a toujours:

$$\forall p \in P \text{ et } t \in T : Pre(p,t) \leq 1 \text{ et } Post(p,t) \leq 1$$

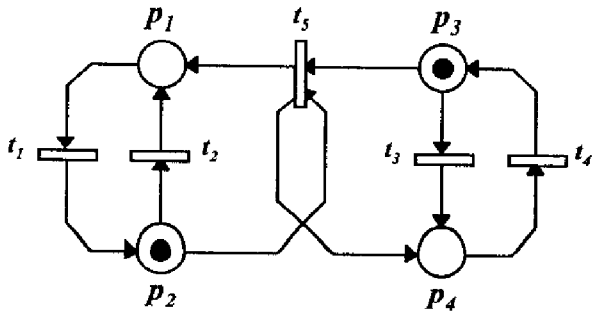


Figure III.1: Un Exemple de Graphe Associé à un Réseau de Petri

La représentation algébrique de ce réseau est donnée par:

- $P = \{p_1, p_2, p_3, p_4\}$ et $|P| = 4$

- $T=\{t_1, t_2, t_3, t_4, t_5\}$ et $|T|=5$
- $Pre=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ et $Post=\begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$
- $C=\begin{bmatrix} -1 & 1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 \end{bmatrix}$ et $M=\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$

L'ensemble des places amonts de t_5 se note: ${}^{\circ}t_5=\{p_2, p_3\}$. De même, l'ensemble des transitions avales de p_1 s'écrit: $p_1^{\circ}=\{t_1\}$.

II.2. Evolution

II.2.1. Règles d'Evolution

L'état d'un réseau de Petri est donné par le marquage de ce réseau, c'est à dire la répartition des jetons dans ses différentes places. L'ensemble des marquages qu'il peut atteindre au cours de son évolution décrit donc l'ensemble des états d'un réseau. Cette évolution, est dictée par deux règles essentielles associées aux transitions. Une règle pour vérifier les conditions d'évolution ou pré-conditions (*transition franchissable*) et une règle pour décrire cette évolution ou post-conditions (*franchissement de transition*).

Définition III.2 ***Transition Franchissable***

Soit M le marquage d'un réseau marqué N, une transition t est *franchissable* (ou sensibilisée ou validée) si et seulement si:

$$\forall p \in P \quad M(p) \geq Pre(p,t)$$

On peut également exprimer le fait que t soit *franchissable* par:

$$M \geq Pre(.,t) \quad \text{ou} \quad M(t> \quad \text{ou} \quad M \xrightarrow{t}$$

❑

Graphiquement, on peut traduire cette condition d'évolution par: un réseau marqué N peut évoluer au niveau de t si et seulement si chaque place amont p de t ($p \in {}^\circ t$) contient un nombre de jetons supérieur ou égal au poids de l'arc reliant p à t.

Définition III.3 ***Franchissement d'une Transition***

Soit M le marquage d'un réseau marqué N, si t est franchissable pour M, le *franchissement* de t (ou tir) donne le nouveau marquage M' tel que:

$$\forall p \in P \quad M'(p) = M(p) - Pre(p,t) + Post(p,t)$$

On décrit également le *franchissement* de t par:

$$M' = M - Pre(.,t) + Post(.t) \quad \text{ou} \quad M(t> M' \quad \text{ou} \quad M \xrightarrow{t} M'$$

❑

L'évolution d'un réseau, c'est à dire le franchissement d'une transition t, est donc une action indivisible qui consiste à retirer un nombre de jetons dans chaque place p ($p \in {}^\circ t$) égal au poids de l'arc reliant p à t et à ajouter dans chaque place avale p' de t ($p' \in t^\circ$) un nombre de jetons égal au poids de l'arc reliant t à p'. L'évolution du marquage est alors clairement définie.

Définition III.4 ***Séquence de Franchissement***

On appelle *séquence de franchissement* une suite de tirs de transition.
Si $M_1 \xrightarrow{t_a} M_2$ et $M_2 \xrightarrow{t_b} M_3$, on dit que la séquence « t_a t_b » est franchissable à partir de M_1 et donne le marquage M_2. On note:

$$M_1(t_a\ t_b > M_3 \qquad \text{ou} \qquad M_1 \xrightarrow{t_a\ t_b} M_3$$

❑

Une *séquence de franchissement* s est décrite sous forme algébrique par un vecteur appelé *vecteur caractéristique* noté $\vec{s}$. Ce vecteur a pour dimension $|T|$ et chaque composante $s(t)$ donne le nombre d'occurrences de la transition t dans s.

II.2.2. Parallélisme et Conflit

Nous venons de voir que l'évolution d'un réseau de Petri est réalisée par le franchissement de transitions sensibilisées. Pour un marquage donné, plusieurs transitions peuvent être sensibilisées mais la règle d'évolution impose qu'une seule transition puisse être franchie à la fois.
Si les transitions sensibilisées n'ont pas de place amont en commun, le tir de l'une d'entre elles ne remet pas en cause celui des autres. On dit alors que les transitions sont *parallèles*. L'évolution du réseau décrit dans ce cas des processus qui ne sont pas en compétition.
Si, en revanche, les transitions sensibilisées ont des places amonts en commun, le tir de l'une d'entre elles peut remettre en question celui des autres. Ces transitions sont *en conflit* ce qui permet de décrire un indéterminisme dans l'évolution d'un réseau lié à la compétition entre ces transitions.

Définition III.5 ***Parallélisme***

Soit un réseau de Petri marqué $N = \langle R;\ M_0 \rangle$.
Deux transitions t_1 et t_2 sont *parallèles structurellement* si et seulement si elles n'ont aucune place d'entrée commune:

$$\left(Pre(.,t_1)\right)^T \times Pre(.,t_2) = 0$$

Ces transitions sont en *parallélisme effectif* si et seulement si elles sont en *parallélisme structurel* et:

$$M \geq Pre(.,t_1) \qquad \text{et} \qquad M \geq Pre(.,t_2)$$

❑

Définition III.6 ***Conflit***

Soit un réseau de Petri marqué $N = \langle R;\ M_0 \rangle$.

Deux transitions t_1 et t_2 sont en *conflit structurel* si et seulement si elles ont au moins une place d'entrée en commun:

$$\exists p \in P \; Pre(p,t_1).Pre(p,t_2) \neq 0$$

Ces transitions sont en *conflit effectif* pour un marquage M si et seulement si elles sont en *conflit structurel* et que:

$$M \geq Pre(.,t_1) \qquad \text{et} \qquad M \geq Pre(.,t_2)$$

❑

Les notions qui précèdent sont essentielles à la compréhension d'un réseau de Petri. Elles décrivent l'évolution d'un réseau, elles sont simples et très faciles à illustrer graphiquement. C'est cette connaissance minimale que devront posséder les personnes qui cherchent à comprendre la logique d'un réseau de Petri qui leur est proposé pour décrire l'évolution d'un système à la conception/étude duquel ils participent.

II.2.3. Marquages Accessibles et Equation Fondamentale

Nous décrivons maintenant les moyens de caractériser toutes les évolutions possibles d'un réseau, c'est à dire l'ensemble des marquages qu'un réseau peut atteindre.

L'ensemble des marquages effectivement atteignables depuis le marquage initial M_0 est appelé *ensemble des marquages accessibles*.

Définition III.7 ***Ensemble des Marquages Accessibles***

L'*ensemble des marquages accessibles* noté $A(R; M_0)$ est défini par:

$$A(R; M_0) = \left\{ M_i, \; \exists s \; M_0 \xrightarrow{s} M_i \right\}$$

❑

Cet ensemble représente le modèle Etats-Transitions associé à un réseau de Petri. Lorsque $A(R; M_0)$ est fini, on peut donc le représenter sous forme graphique ou matricielle. Le graphe associé, appelé *graphe des marquages* et noté $G(R; M_0)$, a pour sommets l'ensemble des marquages accessibles. Un arc relie alors deux sommets M et M' s'il existe une transition (au sens réseau de Petri) pour passer de M à M'. Chaque arc est étiqueté par le nom de la transition correspondante.

Une autre approche pour caractériser les marquages d'un réseau repose sur sa formulation algébrique qui permet de décrire un ensemble de marquages à partir d'une équation appelée *équation fondamentale*.

Définition III.8 ***Equation Fondamentale***

L'évolution d'un réseau de Petri, à partir d'un marquage M et sous occurrence d'une séquence de franchissement *s*, est décrite par son *équation fondamentale*:

$$M' = M - Pre.\overline{s} + Post.\overline{s}$$

Ou, de façon plus concise à partir de *C*,

$$M' = M + C.\overline{s} \qquad \textbf{(III.1)}$$

❑

Cette équation décrit un ensemble de marquages plus grand que celui des marquages accessibles. Ceci est du à la perte de la notion d'ordre sur les transitions de la séquence *s* dans le vecteur caractéristique $\overline{s}$. Cette approche assure toutefois que, tout marquage qu'un réseau peut atteindre, vérifie l'équation (III.1) et sera particulièrement utile lorsque l'énumération complète des marquages accessibles n'est pas envisageable.

Notations

Soit $\langle R;\ M_0 \rangle$ un Réseau de Petri marqué, si $|P| = n$ alors tout marquage peut s'écrire, comme on l'a vu, par un vecteur colonne de dimension *n*, mais également sous la forme $M = P_1^{\mu_1}.P_2^{\mu_2}...P_n^{\mu_n}$ avec $\mu_i \geq 0$, nombre de jetons contenus par la place P_i..

On note de plus:

- $|M|$: longueur de *M*, c'est à dire le nombre de places marquées pour *M*,
- $|M|_{P_i}$: nombre de jetons dans P_i pour le marquage *M* ($|M|_{P_i} = \mu_i$),
- $im(M) = \{p \in P \ / \ |M|_p > 0\}$: l'ensemble des places marquées

❑

II.2.4. Exemple

Le réseau de Petri présenté dans l'exemple du §III.1.2 permet de décrire une redondance passive simple avec réparation (ou remplacement). Ce réseau est alors interprété comme suit:

- Le sous-réseau engendré par les places p_1 et p_2 correspond à l'élément en mode nominal: actif (*ON*) ou défaillant (*HSon*). La transition t_1 représente la défaillance de l'élément en mode nominal (*Don)* tandis que t_2 décrit sa réparation (*Ron*).
- Le sous-réseau engendré par les places p_3 et p_4 correspond à l'élément en mode secours : en attente (*SB*) ou défaillants (*HSsb*). La transition t_3 représente la défaillance de l'élément en mode secours (*Dsb)* tandis que t_4 décrit sa réparation (*Rsb*).
- Ces deux sous-réseaux sont synchronisés par la transition t_5 qui traduit le changement de mode ou commutation entre un élément nominal défaillant et un élément de secours en attente (*C*).
- Le marquage initial indique qu'il y a un élément nominal défaillant (un jeton dans p_2) et un élément de secours en attente (un jeton dans p_3).

La Figure III.2 illustre une telle interprétation.

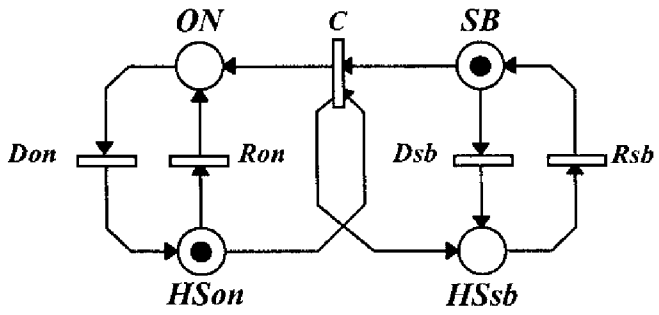


Figure III.2 : Une Interprétation Possible du Réseau de la Figure III.1

Pour le marquage donné (jetons noirs), l'ensemble des *transitions franchissables* est $\{Ron, Dsb, C\}$.

Les transitions *Ron* et *Dsb* sont en *parallélisme effectif* tandis que les transitions *Ron* et *C* sont en *conflit effectif.*

Le *franchissement de la transition C* va supprimer un jeton dans chacune des places *HSon* et *SB* et générer un jeton dans chacune des places *ON* et *HSsb*. On obtient le marquage décrit par les jetons gris.

La *séquence de franchissement s* = « *C Rsb Don* », à partir du marquage noir, ramène le réseau dans ce même marquage.

$$M \xrightarrow{s} M$$

Cette séquence correspond à la commutation entre un élément nominal défaillant et un élément de secours en attente, suivi de la réparation de l'élément de secours et de la défaillance de l'élément nominal actif.

L'équation fondamentale du réseau s'écrit :

$$M' = M + \begin{bmatrix} -1 & 1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 \end{bmatrix} \times \overline{s}$$

On vérifie l'invariance du marquage par le franchissement de *s* = « *C Rsb Don* » à partir du marquage noir. On a:

$$M = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad \text{et} \quad \overline{s} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

On obtient donc, à partir de l'équation fondamentale, le marquage M' tel que:

$$M' = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 1 & -1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = M$$

Le *graphe des marquages accessibles* est donné par la Figure III.3. Le marquage initial M_0 correspond au sommet en gris. Ce graphe représente donc le modèle Etats-Transitions que l'on aurait pu construire directement pour une redondance passive avec réparation.

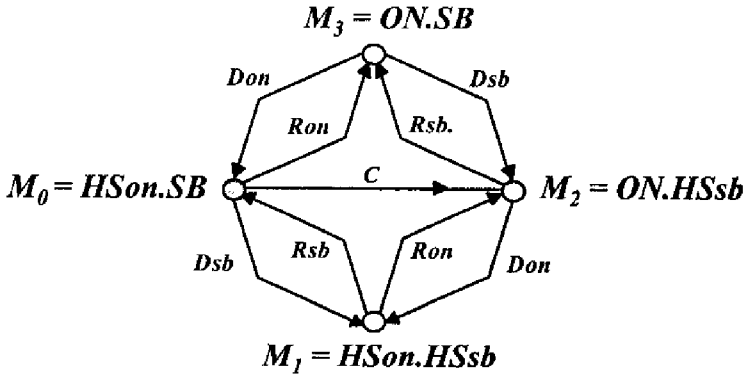


Figure III.3: Graphe des Marquages Accessibles

II.3. Propriétés

II.3.1. « Bonnes Propriétés »

On regroupe sous ce terme l'ensemble des propriétés liées aux évolutions possibles d'un réseau de Petri à partir d'un marquage initial donné.

Définition III.8 ***Réseau K-borné***

Une place $p \in P$ d'un réseau de Petri marqué $N = \langle R; M_0 \rangle$ est dite *k-bornée*, *k entier naturel*, si et seulement si:

$$\forall M \in A(R; M_0)\ M(p) \leq k$$

Si $k=1$, on dit que la place est *binaire*.

Un réseau de Petri marqué N est *k-borné* si et seulement si toutes ses places sont *k-bornées*. Il est binaire si et seulement si toutes ses places sont binaires.

❑

Un résultat très important permet d'affirmer que cette propriété est décidable. C'est à dire que l'on peut théoriquement déterminer si un réseau marqué est *k-borné* ou non.

Le caractère borné d'un réseau traduit la finitude du nombre d'états du système qu'il décrit. Cette propriété permettra de vérifier l'Hypothèse II.2 formulée pour les modèles de Disponibilité présentés au chapitre précédent.

Définition III.9 ***Réseau Quasi-Vivant***

Une transition $t \in T$ d'un réseau marqué $N = \langle R; M_0 \rangle$ est *quasi-vivante* si et seulement si il existe une séquence de franchissement s qui mène, à partir du marquage initial, dans un marquage tel que t est franchissable. C'est à dire:

$$t \text{ est } \textit{quasi-vivante} \quad \text{ssi} \quad M_0 \xrightarrow{s} M \xrightarrow{t} M'$$

Un réseau est *quasi-vivant* si et seulement si toutes ses transitions sont quasi-vivantes.

❑

La quasi-vivacité d'un réseau traduit le fait qu'à partir d'un état initial, le système qu'il représente a une chance d'évoluer jusqu'à un certain état tel qu'un événement donné peut survenir. Ceci étant vrai pour tous les événements représentés. Cette propriété permet de vérifier, par exemple, que certaines défaillances seront bien prises en compte dans le modèle. Elle sera particulièrement utile dans le cas de systèmes non répétitifs.

Définition III.10 ***Réseau Vivant***

Une transition $t \in T$ d'un réseau marqué $N = \langle R; M_0 \rangle$ est *vivante* si et seulement si, quel que soit un marquage de l'ensemble des marquages accessibles, il existe une séquence de franchissement s qui mène, à partir de ce marquage, dans un marquage tel que t est franchissable. C'est à dire:

$$t \text{ est } \textit{vivante} \quad \text{ssi} \quad \forall M \in A(R; M_0)\ \exists s\ M \xrightarrow{s\,t}$$

Un réseau est *vivant* si et seulement si toutes ses transitions sont vivantes.

❑

On a : Vivacité $\Rightarrow$ Quasi-Vivacité

La vivacité est une propriété plus forte que la quasi-vivacité. Elle indique que, quelle que soit l'évolution d'un réseau, aucune transition ne deviendra infranchissable. Elle permet donc de prouver qu'un système est à évolution permanente (sans blocages), et ce, sur l'ensemble des événements décrits.

Définition III.11 ***Réseau Réinitialisable***

Un réseau marqué $N = \langle R; M_0 \rangle$ est *réinitialisable* si et seulement si, quel que soit le marquage de l'ensemble des marquages accessibles, il existe une séquence de franchissement qui mène dans le marquage initial. C'est à dire:

$$\forall M \in A(R; M_0)\ \exists s\ M \xrightarrow{s} M_0$$

❑

Cette propriété permet de prouver le caractère répétitif d'un système. Elle assure également que ce système est sans blocage, mais contrairement à la vivacité elle ne permet pas de dire que tous les événements décrits pourront intervenir.

Définition III.12 ***Etat d'Accueil***

Un marquage M d'un réseau marqué $N = \langle R; M_0 \rangle$ est *un état d'accueil* si et seulement si quel que soit le marquage de l'ensemble des marquages accessibles, il existe une séquence de franchissement qui mène dans M. C'est à dire:

$$\forall M' \in A(R; M_0)\ \exists s\ M' \xrightarrow{s} M$$

❑

L'état initial d'un réseau réinitialisable est un état d'accueil particulier. Cette propriété met donc également en évidence le caractère répétitif.

II.3.2. Vérification des « Bonnes » Propriétés

La vérification des « bonnes » propriétés fait l'objet d'une active recherche depuis la définition du modèle par Carl Adam Petri. Elle est particulièrement délicate lorsque le réseau n'est pas k-borné. Dans le cas contraire, le réseau a un nombre fini de marquages accessibles et si l'on est capable de générer le graphe associé on peut alors directement vérifier les autres propriétés par l'analyse de ce dernier.

Comme on l'a souligné la propriété *k*-borné est théoriquement décidable. En pratique, on peut la vérifier en appliquant l'algorithme générant *l'arbre de couverture* du réseau marqué. Sans entrer dans les détails, cet algorithme, décrit dans [Peterson 81], permet de générer automatiquement le graphe des marquages accessibles, lorsque le réseau est *k*-borné, ou d'affirmer qu'il n'est pas *k*-borné. Autrement dit, on peut obtenir directement le modèle Etats-Transitions correspondant à un système décrit par un réseau de Petri *k*-borné.

La théorie des graphes offre alors tous les outils nécessaires à l'analyse des autres propriétés. En effet, soit un réseau marqué $N = \langle R;\ M_0 \rangle$:

- Une transition $t \in T$ est quasi-vivante si et seulement si, à partir du sommet initial M_0, il existe au moins un chemin menant de M_0 à un sommet M de $G(R;\ M_0)$ tel qu'un arc étiqueté par t est issu de M.
- Une transition $t \in T$ est vivante si et seulement si, quel que soit un sommet M de $G(R;\ M_0)$, il existe au moins un chemin menant à un sommet M' tel qu'un arc étiqueté par t est issu de M'.
- Si $G(R;\ M_0)$ est fortement connexe le réseau est réinitialisable et tous les marquages sont des états d'accueil.

La forte connexité du graphe est une propriété puissante qui, lorsqu'elle est prouvée permet d'affirmer la vivacité en vérifiant simplement que chaque transition du réseau étiquette au moins un des arcs du graphe.

Nous n'avons présenté ici qu'une seule méthode de vérification des « bonnes » propriétés. Cette méthode repose sur la génération du graphe des marquages accessibles. Elle est très efficace dans la mesure où ce graphe peut être généré (réseau *k*-borné) et que le nombre de sommets n'est pas trop important. Notons cependant que les capacités de mémoire et de calcul des ordinateurs actuels permet de pouvoir générer et analyser des graphes comportant un grand nombre de sommets (plusieurs millions).

Exemple

Il est simple de vérifier sur la Figure III.3 que le graphe est borné, fortement connexe et que chaque transition apparaît. Le réseau décrivant la redondance passive avec réparation possède donc toutes les « bonnes » propriétés. On peut rattacher ces propriétés au comportement du système ainsi modélisé.

- Le réseau est *1*-borné, donc il n'y aura jamais deux éléments actifs ou en attente en même temps.
- Tous les états sont des états d'accueil. Quelle que soit son évolution future, le système pourra toujours revenir dans chacun de ses états.

- Le réseau est vivant. Toutes les défaillances, réparations et commutations restent possibles, il n'y aucune dégradation irréversible.

Ces caractéristiques étant conformes au comportement espéré du système, on a pu, grâce à la vérification des propriétés, apporter une certaine validation du modèle.

Si, dans cet exemple, le système à modéliser était simple et aurait pu l'être directement à partir d'un modèle Etats-Transitions, la vérification des « bonnes » propriétés est particulièrement intéressante pour un réseau de Petri complexe. Elle permet d'augmenter sensiblement, et de façon formelle, la confiance que l'on peut lui porter dans sa capacité à représenter correctement le système que l'on étudie.

□

II.3.3. Propriétés Structurelles

Les « bonnes » propriétés concernent l'évolution d'un réseau de Petri marqué. Nous présentons maintenant les propriétés liées à la structure d'un réseau. Elles s'appuient sur sa formulation algébrique et ne nécessitent pas l'énumération des marquages accessibles.

La structure d'un réseau est décrite par les matrices *Pre* et *Post*, et si ce réseau est pur, elle est également entièrement représentée par la matrice *C*. Cette matrice intervient dans l'équation fondamentale. Rappelons-la:

$$M' = M + C.\overline{s} \qquad \textbf{(III.1)}$$

Les propriétés structurelles sont directement issues de l'étude des noyaux à droite et à gauche de *C*. Ils permettront, à partir de l'équation (III.1), de dégager des ensembles de transitions et de places au comportement singulier.

II.3.3.1 Composantes Répétitives et Invariants de Transitions

Le noyau à droite de la matrice *C* donne un ensemble de vecteurs $\overline{s}$ vérifiant:

$$C.\overline{s} = 0$$

Donc tout vecteur du noyau à droite est tel que le marquage M' obtenu par l'équation fondamentale est identique à *M*. Chacun de ces vecteurs est appelé *composante répétitive stationnaire* (ou *p* semi-flot). Si, sur l'ensemble de ces composantes, toute transition *t* est représentée ($\overline{s}(t) \neq 0$) alors le réseau est dit *répétitif*. Cette propriété est indépendante du marquage initial.

Par ailleurs, s'il existe, pour un réseau de Petri marqué, une séquence franchissable dont le vecteur caractéristique est une composante répétitive positive, alors cette séquence ramène le réseau dans le même marquage. Elle est appelée *invariant de transition*. Un invariant de transition dépend, lui, du marquage initial du réseau.

Le caractère répétitif est une propriété moins forte que la vivacité. En effet, si un réseau est répétitif, cela ne veut pas forcément dire que chaque composante répétitive correspond à des séquences effectivement franchissables. On a en fait :

$$\textit{Vivacité} + \textit{Borné} \quad \Rightarrow \quad \textit{Répétitivité}$$

Toutefois, ne contredisant pas la vivacité, cette propriété sera intéressante à étudier lorsque on ne peut décider directement de la vivacité.

II.3.3.2 Composantes Conservatives et Invariants de Places

On peut prémultiplier chaque membre de l'équation (III.1) par un vecteur de pondération f^T de dimension $1 \times |P|$. On note $f^T(p)$ la composante de f^T qui donne la pondération du marquage associé à une place p. L'équation (III.1) devient:

$$f^T.M' = f^T.M + f^T.C.\vec{s} \qquad \textbf{(III.2)}$$

Le noyau à gauche de la matrice C donne un ensemble de vecteurs f^T vérifiant:

$$f^T.C = 0$$

Ces vecteurs sont appelés *composantes conservatives*. On ne retient, en général, que les solutions positives, ce sont alors des *composantes conservatives positives*.
Si, sur l'ensemble de ces composantes positives, toute place p est représentée ($f^T(p) \neq 0$) alors le réseau est dit *conservatif*. Cette propriété est indépendante du marquage initial.

Pour toute composante conservative, l'équation (III.2) se résume à:

$$f^T.M' = f^T.M = cte = f^T.M_0 \ , \ \forall M \in A(R; M_0) \qquad \textbf{(III.3)}$$

Cette équation décrit une fonction linéaire sur un ensemble de places (celles pour lesquelles $f^T(p) \neq 0$). Elle est appelée *invariant de place*. Elle désigne donc un ensemble de places pour lesquelles le marquage initial est conservé.

Si le réseau est conservatif cela permet de prouver son caractère borné. On dit que le réseau est *structurellement* borné.

$$\textit{Réseau Conservatif} \quad \Rightarrow \quad \textit{Réseau Structurellement Borné}$$

Pour un réseau conservatif avec un marquage initial donné, on peut déterminer, grâce aux invariants de places, une borne pour chacune des places. En effet, pour les composantes conservatives positives, l'équation (III.3) s'écrit également:

$$\sum_{j=1}^{|P|} f^T(p_j).M(p_j) = \sum_{j=1}^{|P|} f^T(p_j).M_0(p_j) \quad avec \quad \forall p,\ f^T(p) \geq 0$$

D'où:

$$f^T(p_i).M(p_i) \leq \sum_{j=1}^{|P|} f^T(p_j).M_0(p_j)$$

Soit:

$$M(p_i) \leq \frac{1}{f^T(p_i)} \sum_{j=1}^{|P|} f^T(p_j).M_0(p_j)$$

La détermination des composantes conservatives et répétitives est automatique à partir de l'étude de la matrice d'incidence *C*. Un algorithme simplifié est présenté dans [Valette 92]. Ces composantes permettent d'obtenir des informations précieuses sur le comportement d'un réseau, indépendamment de son marquage initial. On peut notamment prouver son caractère borné.
D'autre part, pour un marquage donné et si le réseau est à la fois répétitif et conservatif, on peut isoler certains comportements cycliques (invariants de transitions) et donner une borne au marquage de chacune des places, et ce, sans avoir à générer l'ensemble des marquages accessibles.
La recherche de ces propriétés structurelles peut donc être appliquée sur des réseaux complexes comme première étude qualitative et justifier, si le réseau est borné et si le nombre d'états n'est pas trop important, la génération du graphe des marquages accessibles pour une analyse plus fine.
Lorsque la génération de ce graphe est impossible, l'analyse structurelle fournit malgré tout un certain nombre d'informations sur le comportement du réseau.

II.3.3.3 Exemple

Reprenons le réseau de la Figure III.1. Comme on l'a vu, ce réseau permet de décrire la redondance passive simple avec réparation. Il n'est pas plus simple que le modèle Etats-Transitions que l'on aurait pu directement construire. Chaque élément de ce réseau est décrit par un jeton. La position d'un jeton dans les différentes places décrit donc un état possible de l'élément.

Supposons maintenant que l'on veuille représenter la redondance passive *k* parmi *n* avec réparation, quelle que soient les valeurs de *k* et de *n* (avec bien sûr la contrainte $k<n$). L'idée naturelle qui vient est d'augmenter le nombre de jetons dans le réseau pour représenter les *k* éléments en mode nominal et $(n-k)$ éléments en mode secours. En admettant que tous les éléments ne soient pas initialement défaillants, on obtient alors le réseau de la Figure III.4.

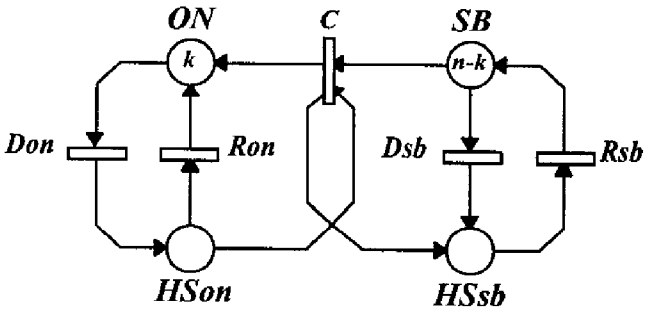


Figure III.4 : Réseau de Petri décrivant la Redondance Passive *k* parmi *n* avec Réparation

Les valeurs de k et n n'étant pas connues à priori ou pouvant être très grandes, on ne peut générer le graphe des marquages accessibles pour vérifier que l'évolution du réseau ainsi augmenté reste conforme au comportement espéré. Le recours à l'analyse des propriétés structurelles s'impose alors.

Le calcul des composantes conservatives positives nous donne les vecteurs de pondération suivants:

$$f_1^T = [1 \quad 1 \quad 0 \quad 0] \quad \text{et} \quad f_2^T = [0 \quad 0 \quad 1 \quad 1]$$

Le réseau est conservatif (ces vecteurs pondèrent toutes les places), il est donc structurellement borné. Le système a toujours un nombre fini d'états.

Pour le marquage initial, les invariants de places nous donnent les équations suivantes:

$$\begin{cases} M(ON) + M(HSon) = k \\ M(SB) + M(HSsb) = n - k \end{cases}$$

On peut alors affirmer qu'il y a toujours k éléments en mode nominal et $(n\text{-}k)$ éléments en mode secours. On peut également dire qu'il y a au plus k éléments nominaux actifs en même temps et $(n\text{-}k)$ éléments de secours en attente en même temps. Ces résultats étant valables quelle que soit l'évolution du modèle.

Parallèlement, le calcul des composantes répétitives positives nous donne les vecteurs caractéristiques suivants:

$$\bar{s}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \bar{s}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \qquad \text{et} \qquad \bar{s}_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Ces vecteurs couvrent l'ensemble des transitions du réseau, ce dernier est donc répétitif. De plus, ils correspondent à des séquences effectivement franchissables à partir du marquage initial. Ces séquences sont les invariants de transitions suivants:

$$\begin{cases} s_1 = Don\ Ron \\ s_2 = Dsb\ Rsb \\ s_3 = Don\ C\ Rsb \end{cases}$$

Ils mettent en évidence les comportements répétitifs suivants:

- Lorsqu'un élément nominal défaille il peut être réparé,
- Lorsqu'un élément en attente défaille il peut être réparé,

- Lorsqu'un élément actif défaille, il peut être remplacé par un élément en attente, puis réparé pour devenir en attente à son tour.

II.3.4. Analyse par Réduction

Nous venons de présenter les techniques d'analyse d'un réseau qui reposent sur la génération du graphe des marquages accessibles (pour les « bonnes » propriétés) et sur l'étude de la matrice d'incidence C (pour les propriétés structurelles). Une troisième approche d'analyse est basée sur des règles de réduction applicables aux noeuds d'un réseau. Ces règles (essentiellement définies dans [Berthelot 86]) permettent de réduire un réseau complexe tout en conservant certaines de ses propriétés. Une fois qu'aucune règle n'est plus applicable, on peut alors étudier le réseau réduit comme décrit précédemment. Cette approche est particulièrement efficace pour les réseaux importants pour lesquels les autres techniques d'analyse peuvent s'avérer lourdes à mettre en oeuvre.

Nous avons choisi de ne pas détailler ces règles (une présentation complète sera trouvée dans [Brams 83]) mais simplement d'en illustrer l'application sur l'exemple qui nous sert de support.

Exemple

Le réseau de la Figure III. nous a permis de décrire la redondance passive k parmi n avec réparation. Ces dernières étaient indépendantes et on disposait d'un nombre de réparateurs toujours suffisant. On impose maintenant au processus de réparation d'être commun pour tous les éléments et avec seulement r réparateurs alloués. De plus, une fois la réparation terminée, on suppose que la mise en place peut échouer.

La Figure III. décrit ce nouveau modèle plus complexe. On retrouve une partie du réseau précédemment décrit. Mais maintenant, lorsqu'un élément est défaillant *HSon* (ou *HSsb*), la transition *DEMon* (ou *DEMsb*) est franchie afin de représenter la demande de réparation. Cette demande est mémorisée dans la place *Mon* (ou *Msb*) et la place *DEM* devient marquée. S'il y a au moins un réparateur disponible (place *Ressources* non vide) alors la réparation est lancée (tir de *Déb Rép*) et dure jusqu'au franchissement de *Fin Rép*. Le réparateur est ensuite libéré et la mise en place peut avoir lieu. Si cette dernière échoue (tir de *Echec*) le processus de réparation doit recommencer (à nouveau marquage de *DEM*). Dans le cas contraire (tir de *Réussite*) l'élément réparé est disponible pour le système (marquage de *OK*). Cet élément est alors alloué au mode nominal ou secours (selon de le marquage de *Mon* et *Msb*) par le tir de la transition *Ron* ou *Rsb*.

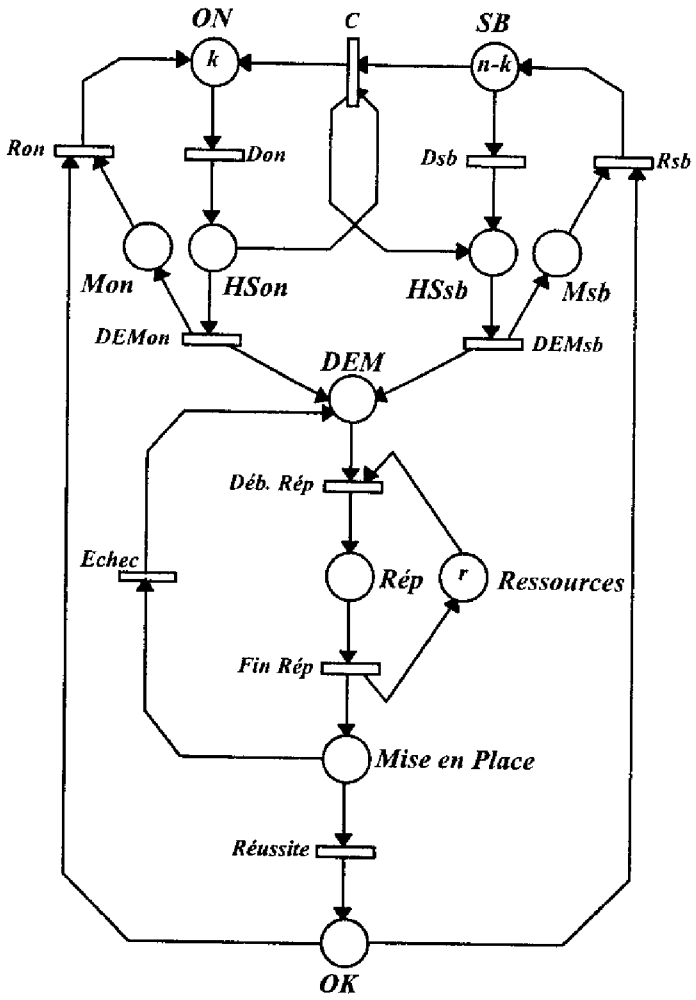


Figure III.5 : Redondance Passive *k* parmi *n* avec Réparation Commune (*r* réparateurs)

Ce réseau est relativement complexe, *n*, *k* et *r* ne sont pas connus à priori, et pour des valeurs très importantes de ces variables, le nombre de marquages accessibles peut être considérable. L'analyse directe par génération du graphe des marquages accessibles n'est donc pas souhaitable.

Nous allons réduire ce réseau pour en diminuer la complexité. Les règles que nous appliquons ici préserve toutes les « bonnes » propriétés, c'est à dire que, si elles sont vérifiées sur le réseau réduit, elles le sont également sur le réseau initial (celui de la figure précédente).

La Figure III.6 montre la réduction du réseau sur la partie qui correspond au processus de remplacement.

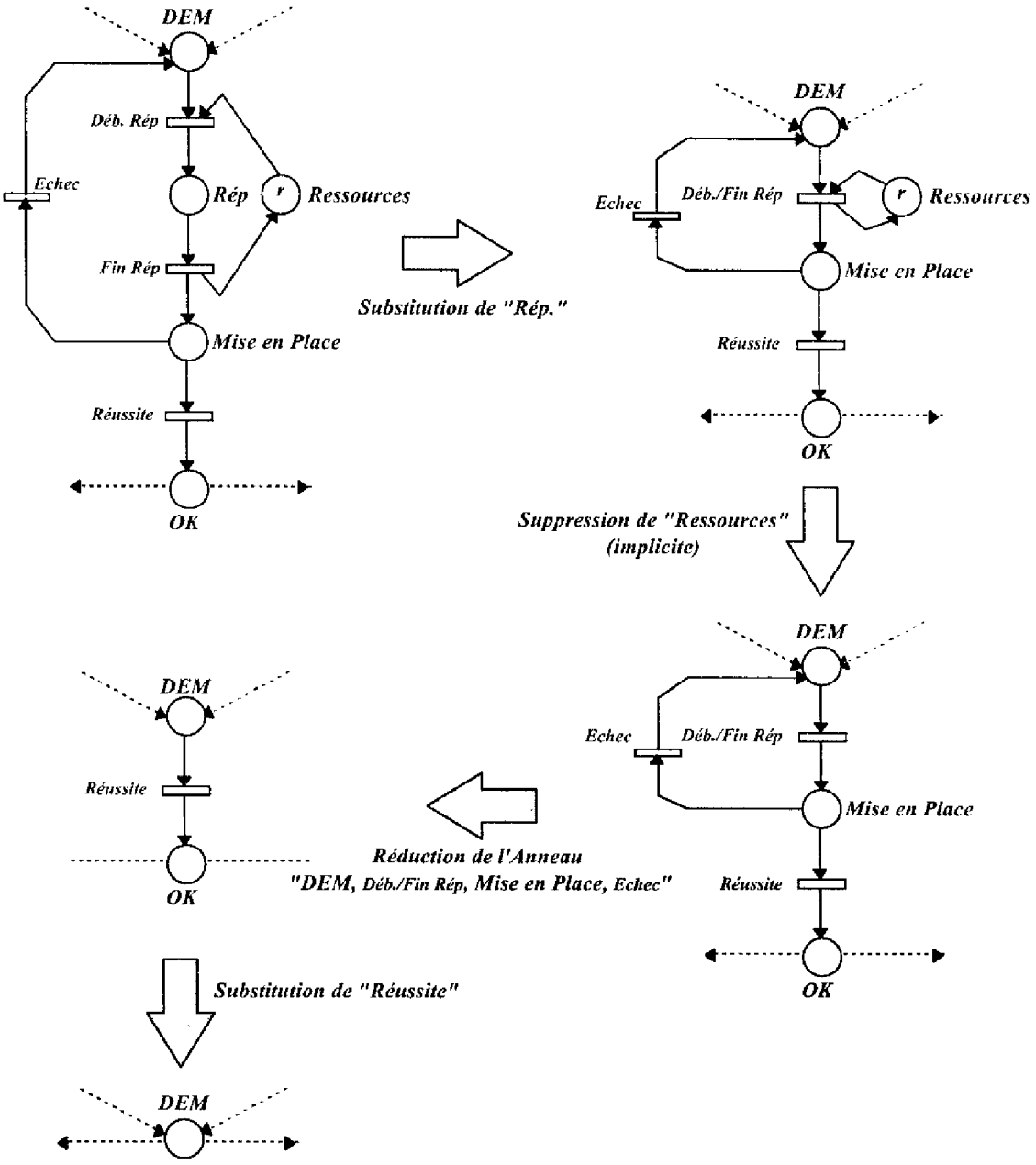


Figure III.6 : Réduction du Processus de Remplacement

On peut alors appliquer de nouveau des règles de réduction sur le réseau ainsi obtenu.

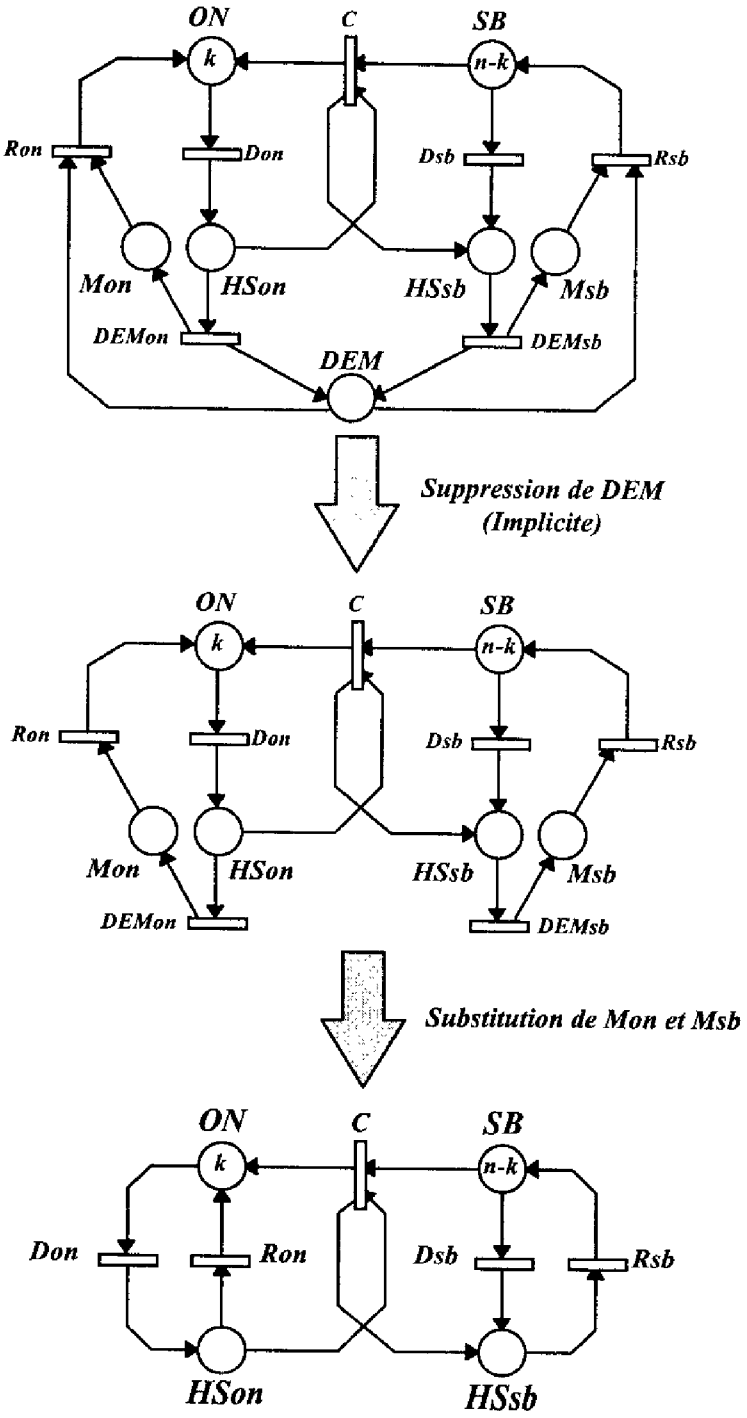


Figure III.7 : Deuxième Partie de la Réduction du Réseau de la Figure III.5

Le réseau réduit finalement obtenu est identique à celui déjà étudier. Si on fixe n et k, on peut vérifier en générant le graphe des marquages accessibles qu'il possède toutes les « bonnes » propriétés donc que le réseau initial les possède également. Le système ainsi modélisé a donc un nombre d'états fini, il n'y a pas de dégradations irréversibles et tous les événements représentés peuvent se produire au cours de son évolution.

Cependant, les règles de réduction ne préservent pas le sens que l'on associe à un réseau. Par exemple, le fait que la réparation soit commune, fasse intervenir un nombre de réparateurs fini et que la mise en place puisse échouer n'est plus apparent sur le réseau réduit obtenu.

❑

II.4. Réseaux de Petri Etendus

Dans ce qui précède, nous avons présenté les réseaux de Petri Généralisés: leur définition, les principes d'évolution ainsi que les propriétés caractéristiques. Ce modèle offre un fort pouvoir d'expression et permet de décrire un grand nombre de comportements inhérents aux systèmes distribués. Cependant certaines situations restent difficiles voire impossibles à décrire:

- Les conditions d'évolution d'un réseau sont basées sur la présence d'un certain nombre de jetons dans les places amonts des transitions mais parfois il peut être intéressant de tester le manque de jetons dans certaines places. Ceci n'est pas directement possible avec la règle d'évolution définie, c'est pourquoi la notion *d'arc inhibiteur* a été introduite.
- Le choix entre plusieurs transitions en conflit effectif n'est soumis à aucune règle. On représente ainsi un indéterminisme qui peut être inhérent au système décrit, mais on peut, dans certains cas, vouloir résoudre cet aléa et imposer un choix entre les transitions en conflit. Ceci est possible en fixant au préalable une règle de priorité entre ces transitions.

Les *réseaux de Petri Etendus* sont des réseaux de Petri Généralisés pour lesquels on a pris en compte la possibilité de décrire ces types de comportements.

II.4.1. Définition et Evolution

Définition III.13 ***Réseau de Petri Etendu [Ciardo 93]***

Un *réseau de Petri Etendu Marqué* est un triplet

$$R_E = \langle R,\ D,\ > \rangle$$

où:

- R est un réseau de Petri Généralisé.
- D: $P \times T \rightarrow N$ est l'application qui à tout couple $(p,\ t)$ associe le poids de l'arc inhibiteur reliant la place p à la transition t.
- $>$ est une relation de priorité acyclique entre les transitions.

Un réseau de Petri Etendu muni du marquage initial M_0 est noté:

$$N_E = (R_E;\ M_0)$$

❑

Les arcs inhibiteurs sont représentés graphiquement comme des arcs amonts mais dont l'extrémité (au niveau de la transition) est représentée par un cercle. Chaque transition intervenant dans une relation de priorité peut être étiquetée par cette relation.

Une transition t d'un réseau de Petri Etendu N_e est franchissable pour le marquage M si elle est franchissable pour le réseau de Petri Généralisé sous-jacent N et si:

- $\forall p \in P,\ D(p,t) > M(p)$ *ou* $D(p,t) = 0$
- il n'existe aucune transition $s \in T$ satisfaisant les conditions précédentes et telle que $s > t$

Le franchissement de la transition t est alors identique à celui défini pour les réseaux de Petri Généralisés.

Remarque Importante

Si les réseaux de Petri Etendus augmentent le pouvoir de modélisation des réseaux de Petri Généralisés, les propriétés présentées précédemment deviennent indécidables. Il n'existe pas d'algorithme permettant notamment de prouver directement qu'un réseau Etendu est borné ou non.
Toutefois, les arcs inhibiteurs et les priorités entre les transitions apportent, par rapport aux réseaux de Petri Généralisés, des conditions supplémentaires pour le franchissement des transitions. Autrement dit, les possibilités d'évolution d'un réseau Etendu sont plus restreintes que celles du réseau Généralisé sous-jacent, donc son ensemble des marquages est inclus dans celui du réseau sous-jacent. Soit:

$$A(R_E;\ M_0) \subseteq A(R;\ M_0)$$

Donc, si le réseau sous-jacent N est k-borné, tout réseau Etendu N_e construit à partir de lui l'est également. On peut alors générer son graphe des marquages accessibles (avec un algorithme prenant en compte ses conditions particulières d'évolution) et étudier directement les autres propriétés.

❑

II.4.2. Exemple

Reprenons le réseau de la redondance passive k parmi n avec réparation commune (Figure III.5). Trois ensembles de transitions sont en conflit structurel:

- { *Echec, Réussite* } représente le résultat de la mise en place de l'élément réparé.
- $\{DEMon,\ C\}$ représente le choix entre l'action de maintenance: soit l'élément nominal défaillant est réparé (tir de DEMon) soit on utilise un élément en attente pour le remplacer (tir de C).
- $\{Ron,\ Rsb\}$ traduit la possibilité d'affecter l'élément réparé au mode nominal (tir de *Ron*) ou au mode attente (tir de *Rsb*), lorsque évidemment les deux modes sont déficients (places *Mon* et *Msb* marquées).

Sur chacun de ces ensembles, le choix de la transition à franchir est aléatoire. Si, pour le premier ensemble, cet aléa est légitime car on veut représenter l'incertitude liée à la mise en place de l'élément réparé, en revanche, il peut être intéressant de fixer ce choix pour les deux autres ensembles. En effet, il semble naturel de privilégier la commutation sur un élément en attente (deuxième ensemble) et de rendre prioritaire le remplacement d'un élément nominal sur celui d'un élément en attente. On peut fixer ces choix grâce aux arcs inhibiteurs et aux priorités entre transitions des réseaux de Petri Etendus. La Figure III.8 montre l'extension du réseau choisie.

Sur le deuxième ensemble, l'arc inhibiteur testant le marquage à zéro de la place *SB* permet de franchir la transition *DEMon* que lorsque *SB* est vide. Ainsi une réparation d'un élément nominal n'intervient que lorsqu'il n'y a plus d'éléments en attente prêt à prendre le relais.

Sur le troisième ensemble, en rendant *Ron* plus prioritaire que *Rsb* ($Ron > Rsb$) le conflit est également résolu. Tout élément nouvellement réparé est donc prioritairement affecté au mode nominal.

Il aurait également été possible de résoudre le deuxième ensemble en utilisant les priorités
($C > DEMon$) et le troisième ensemble en testant sur la transition *Rsb* le marquage à zéro (arc inhibiteur) de la place *Mon*.

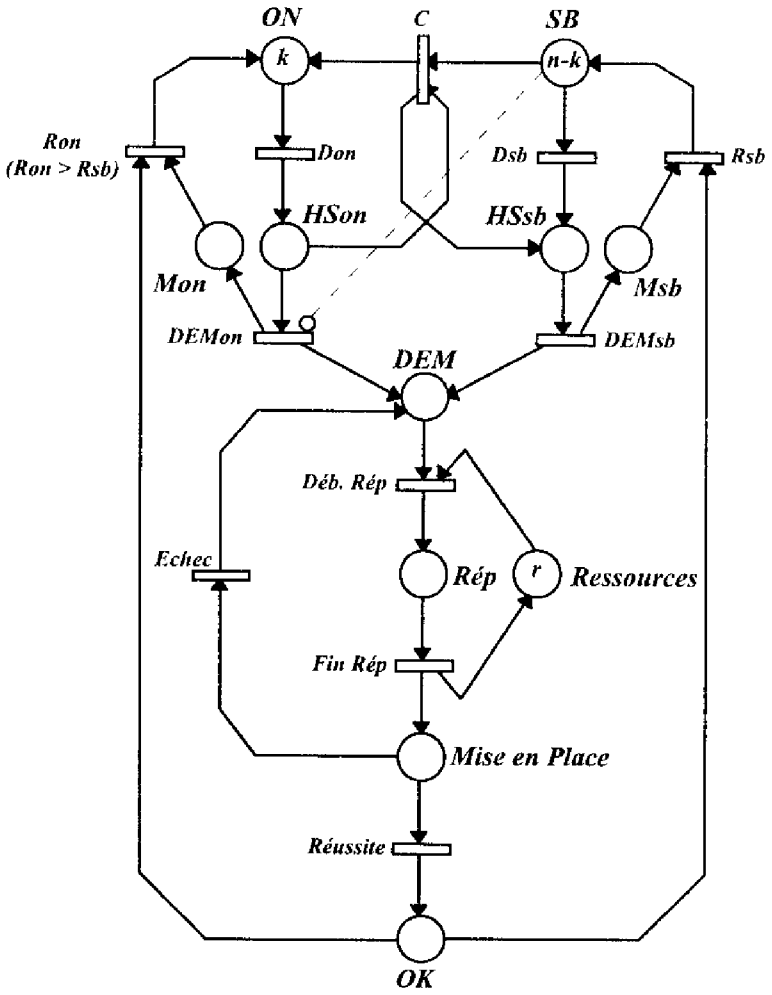


Figure III.8 : Exemple de Réseau de Petri Etendu

Ayant déjà réalisé l'analyse qualitative du réseau sous-jacent, on est en mesure de mieux mesurer l'impact des extensions apportées. Si toutefois des doutes subsistent quant au bon comportement du modèle, on peut toujours générer son graphe des marquages accessibles pour k, n et r fixés (car le réseau sous-jacent est borné) et vérifier les autres propriétés. En revanche les règles de réduction ne sont plus applicables.

Ayant déjà réalisé l'analyse qualitative du réseau sous-jacent, on est en mesure de mieux mesurer l'impact des extensions apportées. Si toutefois des doutes subsistent quant au bon comportement du modèle, on peut toujours générer son graphe des marquages accessibles pour k, n et r fixés (car le réseau sous-jacent est borné) et vérifier les autres propriétés. En revanche les règles de réduction ne sont plus applicables.

II.5. Conclusions

Dans ce chapitre nous avons présenté quelques notions essentielles sur les réseaux de Petri. Pour cela, nous nous sommes appuyés sur un exemple familier des fiabilistes: celui de la redondance passive k parmi n avec réparation. Nous avons pu ainsi illustrer le pouvoir de modélisation des systèmes dynamiques discrets et les possibilités de vérification qu'ils apportent.
Les réseaux de Petri permettent de décrire, tout en maîtrisant l'explosion combinatoire des modèles, des comportements complexes inhérents aux systèmes distribués: le parallélisme, le choix, la compétition et la synchronisation. Si on revient sur l'exemple de la Figure III.5, on a pu représenter le parallélisme entre les défaillances de chaque élément, la compétition pour la prise de la ressource *Réparateurs*, le choix pour décrire les possibilités d'échec à la mise en place ainsi que la synchronisation pour la commutation entre un élément défaillant en mode nominal et un élément de secours en attente. Et ces comportements ont été décrits quelles que soient les valeurs de k, n et r sur un modèle structurellement identique et facilement intelligible grâce à une représentation graphique efficace.
Trois méthodes d'analyse des propriétés d'un réseau ont été présentées: l'analyse par énumération qui est basée sur la génération des marquages accessibles d'un réseau marqué, l'analyse structurelle et enfin l'analyse par réduction. L'étude des propriétés d'un réseau permet une vérification formelle des modèles et ainsi, de mettre à jour des erreurs de modélisation ou d'interprétation des spécifications du système à décrire.
Enfin, nous avons introduit une extension des réseaux de Petri: les réseaux de Petri Etendus qui permettent d'augmenter le pouvoir de modélisation mais au détriment de l'analyse des propriétés.

On peut ainsi dégager une démarche souhaitable à suivre lors de la conception d'un modèle.

1. La première étape consiste à construire un réseau *non Etendu* décrivant la logique d'évolution d'un système donné.
2. On vérifie ensuite les propriétés de ce réseau. On commence par le calcul des invariants de place et de transitions, puis, si le réseau est complexe, on peut appliquer les règles de réduction. Enfin, lorsque c'est possible, on génère le graphe des marquages accessibles pour une analyse plus fine. Si l'analyse des propriétés révèle des erreurs, on reprend la modélisation (étape 1).
3. Lorsque l'analyse du modèle est satisfaisante, on peut éventuellement augmenter le réseau d'arcs inhibiteurs et de priorités entre les transitions pour affiner son comportement. En appliquant cette extension du réseau sur un modèle préalablement vérifié, on est en mesure de mieux en estimer l'impact sur le comportement du modèle. Et si le réseau sous-jacent est borné on peut à nouveau étudier les propriétés du réseau Etendu sur son graphe des marquages accessibles.

Les modèles de réseau de Petri présentés (Généralisés et Etendus) apportent donc des possibilités de modélisation très intéressantes notamment face aux limitations des modèles classiques présentés au chapitre précédent. Ces modèles représentent la logique d'évolution d'un système, mais cette évolution n'est pas quantifiée: les dates de franchissement des transitions ne sont pas données (ex: on ne sait pas quand auront lieu les défaillances) et les choix entre plusieurs transitions en conflit ne sont pas probabilisés (ex: mise en place des éléments réparés). Pour étudier quantitativement le comportement stochastique d'un système modélisé par réseau de Petri, il faut définir explicitement comment évolue le réseau dans le

temps et comment sont pris en compte les aléas liés à son évolution. De telles possibilités font l'objet des extensions temporelles et stochastiques des réseaux de Petri, regroupées sous le terme générique de *Réseaux de Petri Stochastiques*.

Chapitre IV

Les Réseaux de Petri

Stochastiques

Natkin [Natkin 80] et Molloy ([Molloy 81], [Molloy 82]) ont été les premiers chercheurs qui, séparément, ont introduit les réseaux de Petri Stochastiques. Depuis, de très nombreux modèles ont été définis, des synthèses complètes sont proposées dans [Florin 90] et [Ciardo 93].

Ce chapitre, présente les concepts de base liés aux réseaux de Petri Stochastiques puis définit le modèle que nous avons retenu pour nos études. Les possibilités alors offertes pour l'analyse probabiliste et statistique sont ensuite présentées.

II.1. Concepts Liés aux Réseaux de Petri Stochastiques

Le principal objectif des réseaux de Petri Stochastiques est d'associer à la description logique d'un système par réseaux de Petri, la prise en compte de contraintes temporelles et stochastiques. En effet, les modèles précédemment définis ne font pas intervenir le temps de manière explicite, ils décrivent simplement sous quelles conditions et de quelles façons peut évoluer un réseau.

La quantification des dates et des aléas d'évolution soulève une série de problèmes qu'il convient de clarifier. Nous exposons ici les concepts associés dans le cas particulier où la notion de temps est associée aux transitions d'un réseau. C'est en effet la situation la plus généralement rencontrée.

II.1.1. Durées Associées aux Transitions

A chaque transition d'un réseau, un réseau de Petri Stochastique associe une variable aléatoire $\tau(t)$ décrite par une distribution à priori quelconque mais donnée pour t. Cette variable correspond à la durée totale de sensibilisation de la transition nécessaire avant franchissement, lorsque cette dernière est isolée du réseau. C'est à dire qu'une épreuve de cette variable fournirait la date de tir de la transition associée si on gelait toute autre évolution du réseau.

La donnée d'une telle variable ne suffit donc pas à caractériser l'évolution globale d'un réseau. Pour cela, il convient d'éclairer deux points fondamentaux :

- Comment choisir, parmi les transitions sensibilisées par un marquage donné, la prochaine transition à franchir ?
- Comment est affectée la durée de sensibilisation d'une transition en fonction de l'évolution du réseau ?

Cela revient à définir une politique d'évolution et une politique de mémoire.

II.1.2. Politique d'Evolution

Pour un marquage M donné, plusieurs transitions d'un réseau de Petri peuvent être sensibilisées. Il existe, dans ce cas, deux façons de choisir la prochaine transition à franchir.

Modèle Concurrentiel

La première solution consiste à sélectionner la transition dont la durée de sensibilisation est la plus faible. Ce choix correspond au *modèle Concurrentiel* (ou « *Race Model* ») et suppose que les jetons d'un réseau restent toujours disponibles pour chacune de ses transitions.

Soit $T(M)$ l'ensemble des transitions sensibilisées par le marquage M.

$$T(M) = \left\{ t \in T,\ M \xrightarrow{t} \right\}$$

On note a_t la durée de sensibilisation d'une transition t qui dépend de l'évolution globale du réseau. A priori, a_t est différente de $\tau(t)$. On peut définir l'ensemble T_{min} des transitions de $T(M)$ ayant une durée de sensibilisation a_t minimale.

$$T_{min} = \left\{ t \in T(M),\ a_t = \min_{t' \in T(M)} (a_{t'}) \right\}$$

Si $T_{min} = \{t\}$, t est donc la transition choisie. Le choix de la prochaine transition à franchir est donc parfaitement déterminé lorsqu'il n'y a qu'une transition de durée de sensibilisation minimale (c'est le cas lorsque les distributions de $\tau(t)$ sont continues).

Dans le cas contraire, c'est à dire lorsque T_{min} n'est pas un singleton, ce critère de sélection ne suffit plus.

Cette politique suppose que le temps s'écoule de la même manière pour toutes les transitions sensibilisées et permet de décrire l'évolution naturelle d'un système distribué. En revanche elle ne donne pas d'indications sur le choix d'évolution du système lorsque plusieurs processus se terminent en même temps.

Présélection

La seconde politique d'évolution, appelée *modèle de Présélection*, consiste, comme son nom l'indique, à choisir d'abord sur *T(M)* la transition *t* à franchir, puis à considérer sa durée de sensibilisation. Ce choix est réalisé grâce à un tirage aléatoire sur *T(M)*, les transitions de *T(M)* ayant été, au préalable, affectées d'une distribution discrète de probabilité $(\varpi_t)_{t \in T(M)}$ normalisée ($\sum_{t \in T(M)} \varpi_t = 1$).

Cette deuxième politique est très lourde à mettre en œuvre car il faut connaître à priori toutes les configurations de *T(M)* pour tous les marquages du réseau afin d'affecter les distributions discrètes de probabilités. Son principal intérêt est de permettre une sélection explicite entre les transitions.

Solution Retenue

La politique d'exécution que nous retiendrons est celle du modèle concurrentiel. Toutefois, on a vu que si plusieurs transitions franchissables ont une même durée de sensibilisation (transitions de T_{min}) ce modèle ne permet plus de sélectionner la transition à franchir. On effectue alors, par défaut, un tirage aléatoire uniforme sur T_{min}.

Mais, si dans T_{min} il existe des transitions en conflits, il peut être intéressant de probabiliser ce conflit (c'est le cas, par exemple, pour probabiliser l'échec de la mise en place d'un élément réparé cf. Figure III.5). On associe alors sur T_{min}, comme dans le cas de la politique de Présélection, une distribution discrète de probabilité $(\varpi_t)_{t \in T_{min}}$ normalisée $\sum_{t \in T_{min}} \varpi_t = 1$.

II.1.3. Politique de Mémoire

Nous avons vu comment choisir une transition à franchir en fonction des durées de sensibilisation associées à chaque transitions franchissables. Le problème est maintenant de caractériser ces durées en fonction de l'histoire passée de l'évolution du réseau, c'est à dire définir une politique de mémoire. Pour une politique de Présélection ce choix est implicite dans la mesure où l'on ne détermine la date de tir d'une transition que lorsque cette dernière a été sélectionnée. La transition est alors la seule à pouvoir être franchie et elle le sera au bout d'une durée correspondant à une réalisation de la variable aléatoire $\tau(t)$ associée. En revanche, il est nécessaire de définir une politique de mémoire pour le modèle concurrentiel.

Trois politiques ont été introduites. La durée totale de sensibilisation d'une transition *t* ($\tau(t)$) est dans tous les cas déterminée à chaque nouvelle sensibilisation de *t*.

Age Memory

Pour cette politique, la durée de sensibilisation restante a_t d'une transition t, avant franchissement possible, est égale à la durée totale déterminée $\tau(t)$ moins la somme des durées de sensibilisation de t déjà écoulées depuis son dernier tir ou depuis sa première validation. Cette politique est très rarement choisie car elle est très lourde à mettre en œuvre et conduit à un processus stochastique sous-jacent particulièrement complexe à analyser.

Resampling

Dans ce cas, il y a perte des durées de sensibilisation écoulées de toutes les transitions à chaque franchissement. Ainsi, après un franchissement de transition, la durée de sensibilisation restante a_t de chaque transition t est à nouveau calculée et est donc égale à une épreuve de $\tau(t)$. Il est alors impossible de représenter des processus s'effectuant en parallèle dans le temps.

Enabling Memory

Cette dernière politique est un compromis des deux précédentes c'est à dire que les durées de sensibilisation écoulées d'une transition t sont mémorisées tant que t est franchissable. Dès qu'elle ne l'est plus, une nouvelle épreuve de $\tau(t)$ doit être déterminée à sa prochaine validation.

C'est cette dernière politique de mémoire que nous retiendrons. Notons toutefois que lorsque la variable $\tau(t)$ d'une transition t est décrite par une loi « sans mémoire » (continue exponentielle ou discrète géométrique ou dirac nul) les trois politiques de mémoire sont équivalentes pour t.

II.2. Modèle Retenu

II.2.1. Définition

La définition des réseaux de Petri Stochastique que nous proposons ici est basée sur celle des réseaux de Petri Temporisés Stochastiques définis dans [Juanole 91] et [Atamna 94]. Il s'agit d'une version simplifiée de la définition générale des Réseaux de Petri Stochastiques [Ciardo 93 (2)] pour laquelle on considère les politiques d'évolution et de mémoire sélectionnées ci-avant.

Définition IV.1 ***Réseau de Petri Stochastique (RdPS)***

Un *réseau de Petri Stochastique(RdPS)* est un quadruplé

$$R_S = \langle R_E,\ I_0,\ F_0,\ \Omega \rangle$$

où:

- $R_E = \langle R,\ D,\ > \rangle$ est un réseau de Petri Etendu
- I_0 est une fonction intervalle de tir initial qui, à chaque transition $t \in T$, fait correspondre un intervalle de temps de R^+:

$$I_0(t) = [\theta_{min}(t),\ \theta_{max}(t)]$$

  $\theta_{min}(t)$ et $\theta_{max}(t)$ étant des dates relatives à l'instant de validation de t.
- F_0 est une fonction densité de probabilité qui, à chaque transition $t \in T$, fait correspondre une densité de probabilité f_t définie sur l'intervalle $I_0(t)$ et telle que:

$$\int_{\theta_{min}(t)}^{\theta_{max}(t)} f_t(x).dx = 1$$

- $\Omega = \{\varpi_{t,S},\ t \in T,\ S \subset T\ /\ \varpi_{t,S}: A(R_S,\ M_0) \rightarrow R^+\}$ est un ensemble de poids, pouvant dépendre du marquage, associés à t lorsque les transitions de S sont franchissables.

Un *réseau de Petri Stochastique* muni du marquage initial M_0 est noté:

$$N_S = (R_S;\ M_0)$$

❑

La fonction $I_0(t)$ définit un support pour la densité de probabilité associée à la transition t. Lorsque t devient franchissable à la date θ, elle pourra être franchie dans l'intervalle $[\theta + \theta_{min}(t),\ \theta + \theta_{max}(t)]$. Sa date de tir possible est alors déterminée grâce à une épreuve de la variable $\tau(t)$ décrite par f_t. Les distributions les plus courantes sont:

- La distribution exponentielle, notée *Exp(λ)*

$$I_0(t) = [0;\ +\infty[\qquad \text{et} \qquad f_t(x) = \lambda.exp(-\lambda.x)$$

- La distribution uniforme, notée *Unif(a, b)*

$$I_0(t) = [a;\ b] \qquad \text{et} \qquad f_t(x) = \frac{1}{b-a}$$

- La distribution Gaussienne, notée *Gauss(μ, σ)*

$$I_0(t) = [0;\ +\infty[\qquad \text{et} \qquad f_t(x) = \frac{1}{\sqrt{2.\pi.\sigma}} exp\left[-\frac{(x-\mu)^2}{2.\sigma^2}\right]$$

- La distribution discrète de Dirac, notée *Dirac(a)*

$$I_0(t) = [a;\ a] \qquad \text{et} \qquad f_t(x) = \delta(x-a)$$

L'ensemble Ω permet, pour un marquage M, de paramétrer le tirage aléatoire sur l'ensemble T_{min} pour sélectionner la prochaine transition à franchir. Ceci est particulièrement intéressant lorsqu'on veut probabiliser le choix entre plusieurs transitions en conflit. Par défaut on a:

$$\varpi_{t,T_{min}} = \frac{1}{|T_{min}|}$$

Dans la pratique, il est très difficile (voire périlleux) de fixer les coefficients $\varpi_{t,S}$ car il faut connaître, à priori, toutes les configurations de T_{min} pour l'ensemble des marquages accessibles de N_S. Nous verrons les choix effectués pour une mise en œuvre simplifiée.

II.2.2. Algorithme d'Evolution

Nous venons de présenter la définition des réseaux de Petri Stochastique, afin d'en décrire le comportement, nous allons maintenant donner leur algorithme d'évolution.

Soit $R_S = \langle R_E,\ I_0,\ F_0,\ \Omega \rangle$ un réseau de Petri Stochastique. A toute transition $t \in T$ on associe une variable réelle a_t qui décrit la durée de sensibilisation restante pour pouvoir franchir t. On note a le vecteur de ces variables de dimension $|T|$. Initialement (avant le premier pas pour le marquage M_0) on pose : $a_t \leftarrow \infty$. On suppose que le réseau est dans le marquage M_{n-1}, $n \geq 1$.

1. Soit $T(M_{n-1})$ l'ensemble des transitions franchissables déterminées grâce aux règles de validation du réseau de Petri Etendu sous-jacent. Si $T(M_{n-1}) = \varnothing$, M_{n-1} est un marquage puits (plus aucune transition franchissable) et

l'algorithme s'achève. Sinon, $\forall t \in T \setminus T(M_{n-1})$, on pose: $a_t \leftarrow \infty$. Puis, passer à 2.

2. Pour chaque transition $t \in T(M_{n-1})$:
   Si $a_t = \infty$, on calcule une réalisation de la variable aléatoire décrite par $I_0(t)$ et f_t. Soit $\tau(t)$ le résultat on pose $a_t \leftarrow \tau(t)$
   On détermine l'ensemble T_{min} des transitions de $T(M_{n-1})$:

$$T_{min} = \left\{ t \in T(M_{n-1}),\ a_t = \min_{t' \in T(M_{n-1})} (a_{t'}) \right\}$$

   Si $T_{min} = \{t_0\}$, la transition à franchir est connue, passer à 3.
   Sinon, la transition à franchir est sélectionnée par un tirage aléatoire sur T_{min} selon la loi discrète $\left(\varpi_{t,T_{min}}(M_{n-1})\right)_{t \in T_{min}}$ qui par défaut est uniforme. Soit t_0 la transition élue, passer à 3.
3. On effectue les opérations suivantes:

$$\forall t \in T(M_{n-1}) \setminus \{t_0\} \quad a_t \leftarrow a_t - a_{t_0} \qquad \text{et} \qquad a_{t_0} \leftarrow \infty$$

   La transition t_0 est franchie (selon les règles du réseau Généralisé sous-jacent). On recommence alors l'algorithme en 1 avec le nouveau marquage M_n.

Exemple

La figure suivante reprend l'exemple de la redondance passive (cf. Figure III.4) pour lequel on suppose que $k=n=1$. Le réseau décrit est maintenant un réseau Stochastique: les transitions ont été étiquetées de leur fonction de distribution. L'algorithme d'évolution est donné sur la séquence débutant depuis le marquage initial et correspondant à l'invariant de transition suivant: défaillance de l'élément nominal actif, remplacement par l'élément en attente et réparation de l'élément défaillant.

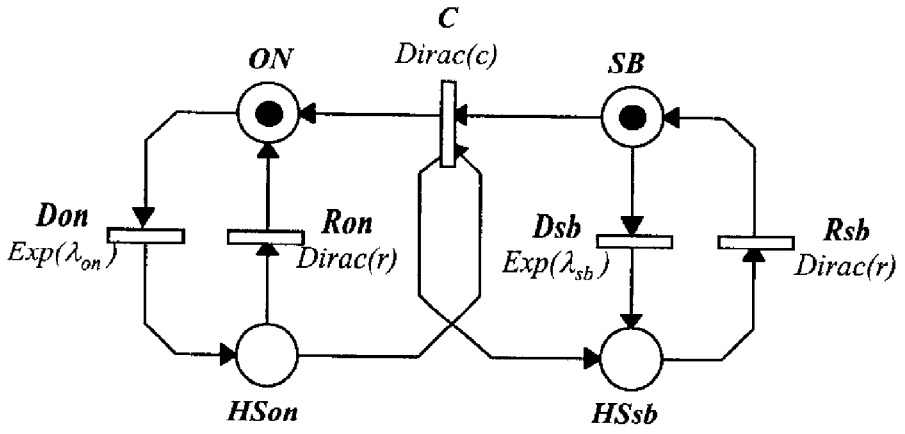


Figure IV.1: Réseau de Petri Stochastique de la Redondance Passive Simple $(k = n = 1)$

On pose: $a_{Don} = a_{Ron} = a_C = a_{Dsb} = a_{Rsb} \leftarrow +\infty$

Pour $M = M_0 = ON.SB$:

1. $T(M_0) = \{Don, Dsb\}$: ces transitions correspondent aux défaillances possibles de l'élément nominal et de l'élément en attente.

2. Comme $a_{Don} = a_{Ron} = +\infty$, on détermine une réalisation de la variable aléatoire $\tau(Don)$ (resp. $\tau(Dsb)$) à partir de la distribution $Exp(\lambda_{on})$ (resp $Exp(\lambda_{sb})$) et on pose: $a_{Don} = \tau(Don)$ et $a_{Dsb} = \tau(Dsb)$. Supposons que $a_{Don} < a_{Ron}$ c'est à dire que l'élément actif est le premier à être défaillant, on a alors: $Tmin = \{Don\}$. *Don* est la prochaine transition à franchir.

3. $a_{Dsb} \leftarrow a_{Dsb} - a_{Don}$ et $a_{Don} \leftarrow +\infty$. *Don* est franchie et on obtient le nouveau marquage $M_1 = HSon.SB$.

Pour $M_1 = HSon.SB$:

1. $T(M_1) = \{Dsb, Ron, C\}$: l'élément en attente peut défaillir et l'élément nominal défaillant peut être réparé ou remplacé par l'élément en attente.

2. Comme $a_{Ron} = a_C = +\infty$, on calcule une réalisation de $\tau(Ron)$ et $\tau(C)$ (à partir des distributions *Dirac(r)* et *Dirac(c)*) et on pose: $a_{Ron} = \tau(Ron)$ et $a_C = \tau(C)$. $a_{Dsb} \neq +\infty$, il reste inchangé. Supposons que l'on a obtenu: $a_C < a_{Ron} < a_{Dsb}$, c'est à dire que le prochain événement sera la commutation entre l'élément nominal défaillant et l'élément actif. Alors $Tmin = \{C\}$, *C* est la prochaine transition à franchir.

3. $a_{Dsb} \leftarrow a_{Dsb} - a_C$, $a_{Ron} \leftarrow a_{Ron} - a_C$ et $a_C \leftarrow +\infty$. *C* est franchie et on obtient le nouveau marquage $M_2 = ON.HSsb$.

Pour $M_2 = ON.HSsb$:

1. $T(M_1) = \{Don, Rsb\}$: l'élément nominal peut défaillir et l'élément en attente peut être réparé. Comme *Dsb* et *Ron* ne sont plus franchissables, on pose: $a_{Dsb} = a_{Ron} \leftarrow +\infty$.

2. Comme $a_{Dson} = a_{Rsb} = +\infty$, on calcule une réalisation de $\tau(Don)$ et $\tau(Rsb)$ (à partir des distributions $Exp(\lambda_{on})$ et *Dirac(r)*) et on pose: $a_{Don} = \tau(Don)$ et $a_{Rsb} = \tau(Rsb)$. Supposons que $a_{Rsb} < a_{Don}$: la réparation de l'élément en attente intervient avant une nouvelle défaillance de l'élément nominal. Alors $Tmin = \{Rsb\}$, *Rsb* est la prochaine transition à franchir.

3. $a_{Don} \leftarrow a_{Don} - a_{Rsb}$ et $a_{Rsb} \leftarrow +\infty$. *Rsb* est franchie et on obtient le nouveau marquage, identique au marquage initial M_0. En revanche la valeur de a est différente de celle du début de l'algorithme (en effet $a_{Don} < +\infty$).

❑

II.2.3. Marquages Accessibles et Etats d'un RdPS

Ensemble des Marquages Accessibles

L'ensemble des marquages accessibles d'un réseau de Petri Stochastique marqué N_S est un sous-ensemble de celui du réseau Etendu N_E sous-jacent. En effet, ce modèle induit des contraintes de franchissement supplémentaires au niveau de chaque transition (contraintes temporelles) qui, pour un marquage donné, peuvent rendre des transitions non franchissables pour R_S alors qu'elles peuvent l'être pour R_E. On a donc:

$$A(R_S; M_0) \subseteq A(R_E; M_0) \subseteq A(R; M_0)$$

Une conséquence de cela est que la finitude de l'ensemble des marquages accessibles du réseau Généralisé sous-jacent N suffit à affirmer celle du réseau Stochastique N_S. Autrement dit si N est borné, N_S l'est également.

Si les lois associées à chaque transition t d'un réseau Stochastique N_S ont toutes pour support $I_0(t) = [0; +\infty[$ alors il s'agit d'un cas particulier pour lequel aucune contrainte de franchissement n'est rajoutée par rapport au réseau Etendu sous-jacent: l'ensemble des marquages accessibles de N_S est alors confondu avec celui de N_E. Donc, les propriétés qualitatives vérifiées par N_E se généralisent alors à N_S.

Etat d'un RdPS

L'état d'un réseau de Petri Stochastique est défini par un couple (M, a) où:

- M est le marquage dans lequel se trouve le réseau
- a est le vecteur de $\left(R^+ \cup \{+\infty\}\right)^{|T|}$ défini pour tout $t \in T$ par:

  si $a_t \in R^+$, c'est le temps pendant lequel la transition devra rester continûment sensibilisée avant de pouvoir être franchie.
  si $a_t = +\infty$, la transition n'est pas sensibilisée par le marquage.

Le vecteur a correspond à celui décrit dans l'algorithme d'évolution. Dans un état (M, a) donné, $min(a_t)$ correspond à la durée restante de séjour du réseau dans le marquage M.

Soit E l'ensemble des états d'un réseau N_S, on classe les états de E en deux sous ensembles: $E = V \cup \Theta$.

- V est l'ensemble des états de durée nulle appelés états *virtuels*

$$V = \{(M,a) \in E \ / \ min(a_t) = 0\}$$

- Θ est l'ensemble des états de durée non nulle appelés états *tangibles*

$$\Theta = \{(M,a) \in E \ / \ min(a_t) > 0\}$$

II.2.4. Transitions Multi-Sensibilisées

Pour des raisons de simplicité de présentation, le modèle réseau de Petri Stochastique défini précédemment ne prend en compte, pour chaque transition, qu'une seule variable aléatoire « durée de sensibilisation » (décrite par le vecteur a). Ceci est très limitatif car, par exemple, sur le réseau de la redondance passive (Figure III.4), il est impossible de considérer plus d'une défaillance à la fois des éléments nominaux. Donc, si une transition est plusieurs fois franchissable, il semble légitime de considérer autant de variables a_t que de façons possibles de franchir cette transition. Mais avant de préciser ce point rappelons la notion de transition plusieurs fois franchissable dite multi-sensibilisée (ou q-validée) grâce à la définition suivante.

Définition IV.1 ***Transition q-validée***

Soit $N_E = (R_E ; M)$ un réseau de Petri Etendu marqué, une transition $t \in T$ est *q-validée* si et seulement si:

$$\min_{p \in {}^\circ t} \left(\left\lfloor \frac{M(p)}{Pré\ (p,t)} \right\rfloor \right) = q ,$$

$q \in N$ et « $\lfloor \ \rfloor$ » représentant la fonction « partie entière »

❑

Ainsi, pour toute transition t q-validée par un marquage M, il faudrait non pas considérer une seule variable a_t donnant la date de tir possible de t, mais q variables a_t pour chacune des façons de prélever *Pré(.,t)* jetons dans les places amonts de t. La mise en œuvre de ce mode de fonctionnement sera présentée dans les paragraphes IV.3. et IV.4.

L'exemple suivant illustre la différence dans la prise en compte du temps entre un réseau considérant la multi-sensibilisation des transitions et l'autre non.

Exemple

On examine le réseau stochastique décrit par la Figure IV.2, on suppose qu'on se place à l'instant initial $\theta = 0$.
La transition t_1 est *3*-validée, en effet on a:

$$\left\lfloor \frac{M(p_1)}{1} \right\rfloor = 4 \text{ et } \quad \left\lfloor \frac{M(p_2)}{2} \right\rfloor = 3$$

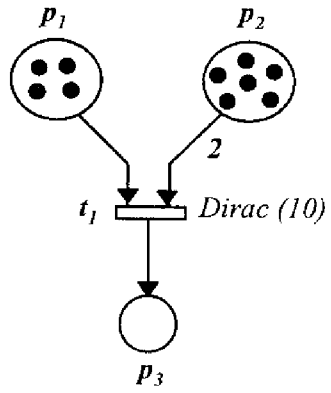


Figure IV.2 : Exemple de Transition q-validée

Si cette multi-sensibilisation n'est pas considérée une seule variable a_{t_1} est prise en compte au premier pas de l'algorithme et, d'après la distribution associée, t_1 sera franchie *3* fois respectivement aux instants $\theta_1 = 10$, $\theta_2 = 20$ et $\theta_3 = 30$.

Dans le cas contraire, trois variables $a_{t_1}(i)$, $i = 1..3$ sont déterminées dès le premier pas, et comme elles ont toutes la même réalisation (variable déterministe), la transition t_1 sera franchie *3* fois à l'instant $\theta_1 = 10$.

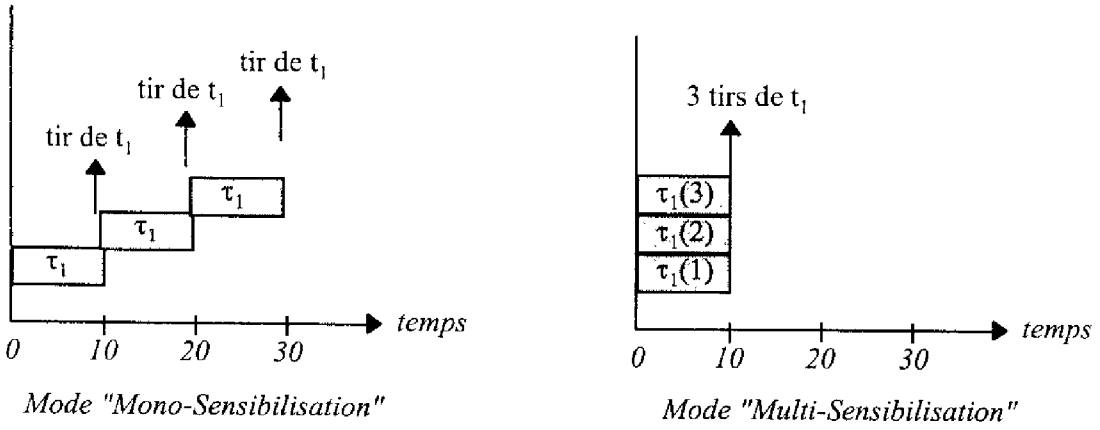


Figure IV.3 : Différence d'Evolution Liée à la Prise en Compte ou non de la Transition *q*-validée

❑

II.1.2. Processus Stochastique Associé

Les réseaux de Petri Stochastiques décrivent plusieurs types de processus Stochastiques [Ciardo 93]. Dans le cadre de nos applications, nous retiendrons exclusivement le processus de marquage à temps continu, $(M_\theta)_{\theta \geq 0}$. Ce processus traduit l'évolution du marquage d'un réseau N_S au cours du temps. Pour pouvoir l'étudier, il ne faut considérer que les marquages

tangibles (ceux liés aux états tangibles) après avoir éliminer les marquages virtuels. L'élimination des marquages virtuels est possible lorsqu'il n'existe pas de suite infinie de marquages virtuels successivement réalisables car sinon l'évolution du système n'est plus basée que sur de tels marquages et l'évolution du modèle est bloquée à la date d'entrée dans de tels marquages. Ceci correspond, dans tous les cas, à des erreurs de modélisation.

Si l'on ne fait aucune hypothèse sur les distributions associées aux transitions, ce processus est très complexe et ne peut généralement pas être étudié par une approche probabiliste. Seule l'approche statistique est envisageable et nécessite donc le recours à la simulation du réseau de Petri Stochastique. Nous présenterons une mise en œuvre possible de cette simulation dans le paragraphe IV.4.

En revanche, si des hypothèses sont faites sur la nature des distributions, il est possible d'évaluer par calcul analytique le processus $(M_\theta)_{\theta \geq 0}$. Ceci est à l'origine de la définition de nombreux modèles dérivés du modèle très général des réseaux de Petri Stochastiques [Ciardo 93 bis]. Parmi eux, celui offrant la plus grand nombre de résultats et qui permet d'exploiter la souplesse du calcul Markovien, est le modèle des réseaux de Petri Stochastiques Généralisés. Nous le présentons dans le paragraphe qui suit.

II.3. Approche Probabiliste

II.3.1. Les Réseaux de Petri Markoviens

Les premières définitions des réseaux de Petri Stochastiques données séparément par Natkin [Natkin 80] et Molloy [Molloy 81] proposent un modèle (maintenant appelé *réseaux de Petri Markoviens*) pour lequel, à chaque transition d'un réseau, est associée une distribution exponentielle. Ainsi, comme on l'a vu, l'ensemble des marquages accessibles d'un tel réseau Stochastique est confondu avec celui du réseau Etendu sous-jacent et les propriétés qualitatives vérifiées par ce dernier ne sont pas remises en cause.

Molloy et Natkin ont montré que le processus de marquage à temps continu présente des propriétés essentielles, car elles permettent de faire la jonction entre les réseaux de Petri et la théorie des processus Markoviens homogènes. Ces propriétés, avec nos propres notations, peuvent s'énoncer comme suit.

Propriétés des Réseaux Markoviens

Si un RdPS N_S d'espace d'état E, est tel que les distributions f_t de chaque transition sont des lois exponentielles, alors $(M_t)_{t \geq 0}$ est *un processus de Markov homogène sur E.* Si le réseau sous-jacent N_E est borné, $(M_t)_{t \geq 0}$ est *à espace d'état E fini* et si N_E est réinitialisable $(M_t)_{t \geq 0}$ est *irréductible* (unicité du régime stationnaire).

❑

Ces résultats sont tout à fait significatifs car ils permettent d'utiliser le pouvoir de modélisation des réseaux de Petri Etendus pour décrire la logique d'évolution d'un système complexe tout en restant compatible avec le calcul Markovien.
Mais pour pouvoir utiliser ce calcul, il est nécessaire de disposer d'une représentation adéquate du processus de marquage $(M_t)_{t \geq 0}$. Cette représentation doit être basée sur un modèle Etats-Transitions, et peut être, comme on l'a vu dans le chapitre II, soit un graphe de Markov, soit une matrice des taux de transition (Générateur Infinitésimal).

Le modèle Etats-Transitions correspondant au processus de marquage d'un réseau borné est directement donné par son graphe des marquages accessibles (cf. III.2.3). Pour obtenir, dans le cas particulier des réseaux Markoviens, le graphe de Markov correspondant, il est donc nécessaire de connaître la valeur des taux de transition entre chaque marquage.
Or, si une unique transition d'un réseau Markovien permet de passer d'un marquage M_i à un autre marquage M_j, alors la probabilité de passer de M_i à M_j est directement donnée par la distribution associée à la transition (du réseau). Donc, comme cette dernière est exponentielle, le taux de transition entre M_i et M_j, correspond au paramètre de cette distribution.
Si plusieurs transitions permettent de passer d'un marquage M_i vers un autre marquage M_j, alors la probabilité de passer de M_i à M_j, est égale à la somme des probabilités de tir associées à chaque transition. Comme les distributions sont exponentielles, le taux de transition entre M_i et M_j, est lui aussi égal à la somme des paramètres de distribution des transitions concernées.

On peut donc, à partir d'un réseau de Petri Markovien borné, générer directement le graphe de Markov (et indirectement le générateur infinitésimal correspondant) du processus de marquage à temps continu. L'étude de ce processus est alors basée, comme dans le cas des

modèles Etats-Transitions, sur la résolution de l'équation différentielle donnant la probabilité instantanée des différents états (cf. chapitre II équation II.5).

II.3.2. Les Réseaux de Petri Stochastiques Généralisés (RdPSG)

Les réseaux de Petri Markoviens, comme on vient de le voir, permettent un accès direct à l'étude probabiliste du processus de marquage d'un réseau de Petri. Cependant, ce modèle est restrictif car toutes les dates d'évolution sont basées sur des distributions exponentielles, et si les taux de transition associés diffèrent de plusieurs ordres de grandeur, on retombe dans le problème des *Stiff Markov Chains* pour la résolution de l'équation II.5. D'autre part, il semble important de pouvoir décrire des synchronisations entre processus qui peuvent être modélisées dans un réseau de Petri par le tir immédiat de transition et non pas après une durée aléatoire exponentiellement répartie. Or, par définition, ceci est impossible avec les réseaux Markoviens.

C'est au vu de ces limitations que les réseaux de Petri Stochastiques Généralisés ont été définis ([Ajmone 84] et [Ajmone 87]). L'idée de ce modèle est d'autoriser deux types de transitions:

- *Les transitions temporisées* basées, comme pour les réseaux Markoviens, sur des distributions exponentielles.
- *Les transitions immédiates* basées sur une distribution de Dirac de durée nulle. Une transition immédiate est franchie dés qu'elle est validée.

Ainsi les synchronisations et les transitions de durée moyenne très faible par rapport aux autres sont décrites par des transitions immédiates.

Un réseau de Petri Stochastique Généralisée présente donc deux types d'état:

- Les états tangibles, pour lesquels l'ensemble des transitions sensibilisées sont des transitions temporisées,
- Les états virtuels, pour lesquels au moins une transition sensibilisée est une transition immédiate.

II.3.2.1 Propriété

Avec ce nouveau modèle une question légitime se pose: L'introduction des transitions immédiates remet-elle en cause le caractère Markovien homogène du processus de marquage ?

Ajmone et al. ont montré que s'il n'existe pas de suite infini d'états virtuels successivement atteignables alors le processus de marquage après élimination des états virtuels reste un processus de Markov homogène. Cette propriété en suivant nos propres notations peut donc s'énoncer ainsi:

Propriété des Réseaux de Petri Stochastiques Généralisés

Si un RdPS N_S, dont $E = V \cup \Theta$ est l'espace d'états (V virtuels, Θ tangibles), est tel que les distributions f_t de chaque transition sont soit des Dirac nuls, soit des lois exponentielles, et si de plus, il n'existe pas une suite infinie d'états virtuels successivement atteignables alors $(M_t)_{t \geq 0}$ est un processus de Markov homogène sur Θ. Si le réseau sous-jacent N_E est borné, $(M_t)_{t \geq 0}$ est *à espace d'état Θ fini* et si N_E est réinitialisable $(M_t)_{t \geq 0}$ est *irréductible* (unicité du régime stationnaire).

❑

L'étude probabiliste d'un RdPSG se basera donc sur celle du processus de marquage lié aux états tangibles décrit par son générateur infinitésimal. Lorsque la condition sur les états virtuels est vérifiée, l'obtention du graphe de Markov des états tangibles est automatique (les algorithmes peuvent être trouvés dans [Ajmone 84] et [Ajmone 87]) et le calcul de la probabilité de chaque état tangible peut être mené.
Nous illustrerons ceci sur l'exemple de la redondance passive.

II.3.2.2 Transitions Multi-Sensibilisées

Le caractère sans mémoire des distributions exponentielles et immédiates permet de prendre en compte simplement la multivalidation des transitions.

En effet, si une transition immédiate t est q-validée à la date θ, le comportement temporel entre le mode mono-sensibilisation et multi-sensibilisation est identique. Il revient au même de franchir 3 fois t respectivement aux instants $\theta_1 = \theta + 0 = \theta$, $\theta_2 = \theta_1 + 0 = \theta + 0 = \theta$ et $\theta_3 = \theta_2 + 0 = \theta + 0 = \theta$ que de franchir 3 fois t à la date $\theta_1 = \theta + 0 = \theta$.

Si d'autre part, une transition temporisée t de paramètre λ est q-validée, alors, pour le mode multi-sensibilisation, la probabilité de franchir t s'écrit:

$$p_t(dt) = \sum_{i=1}^{q} \lambda . dt = (q . \lambda) . dt$$

Autrement dit, on peut simuler le mode multi-validation en considérant un paramètre de loi λ' proportionnel au coefficient q.

II.3.2.3 Transitions Immédiates en Conflit

Nous avons vu dans la définition générale des réseaux de Petri Stochastiques que l'on pouvait probabiliser le choix de la prochaine transition à franchir en affectant une distribution discrète de probabilité $(\omega_{t,T_{min}})$ aux transitions de l'ensemble T_{min}. Pour les RdPSG, cet ensemble est plus grand qu'un singleton uniquement lorsqu'il existe un conflit effectif entre des transitions immédiates. On pourra donc, après avoir examiner l'ensemble des conflits possibles, probabiliser le choix entre les transitions immédiates en fixant au préalable leur coefficient $\omega_{t,T_{min}}$.

II.3.3. Exemple

Reprenons à nouveau l'exemple de la redondance passive k parmi n. On suppose qu'il y a deux éléments actifs et un en attente ($k = 2$ *et* $n = 3$).

Les défaillances et réparations sont des variables aléatoires décrites par une distribution exponentielle. Les transition temporisée *Don* (resp. *Ron*) peut être 2-validée, le paramètre de la distribution est donc fonction du marquage de la place *ON* (resp. HSon).

La commutation s'effectuant en moyenne beaucoup plus rapidement que les défaillances et réparations, la transition C est considérée comme une transition immédiate. Notons qu'une distribution de Dirac nulle peut être assimilée à une distribution exponentielle de taux infini.

La Figure IV.4 décrit ce réseau Stochastique Généralisé. Les transitions immédiates sont représentées par un trait épais et les transitions temporisées par un rectangle.

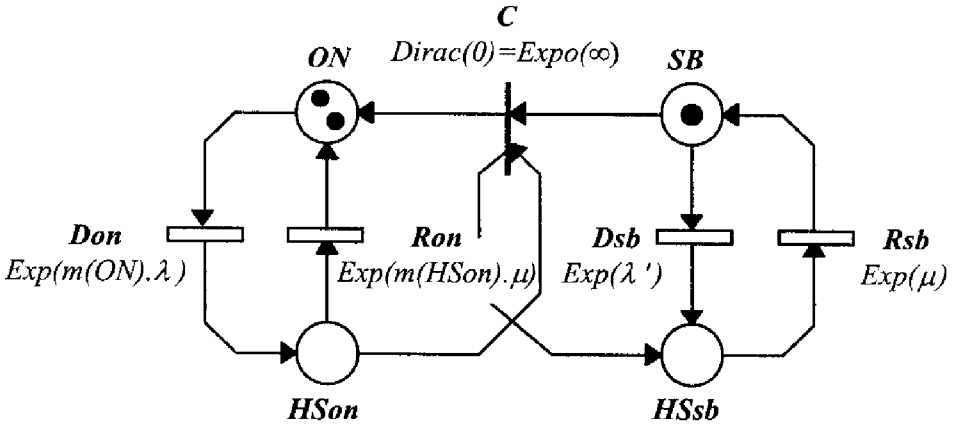


Figure IV.4 : Exemple de RdPSG

Dans cet exemple, il ne peut y avoir de conflit entre transitions immédiates puisque C est l'unique transition de Dirac nul.

La Figure IV.5 donne le graphe des marquages accessibles du réseau Etendu (ici égal au réseau Généralisé) sous-jacent. On peut remarquer que les « bonnes » propriétés, comme dans l'exemple du chapitre précédent, sont vérifiées. Ce réseau est notamment borné et réinitialisable, donc l'espace des états tangibles est borné et le processus de marquage associé admet une unique distribution stationnaire.

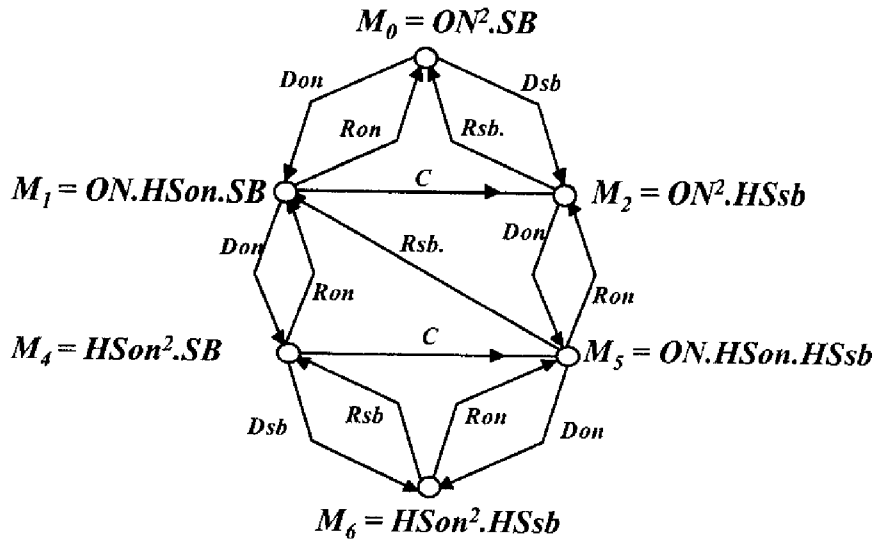


Figure IV.5 : Graphe des Marquages Accessibles

A partir de ce graphe des marquages on peut automatiquement générer le graphe des transitions entre états (Figure IV.5). Lorsqu'un marquage ne sensibilise que des transitions temporisées (marquage tangible), ces taux sont obtenus comme dans le cas des réseaux Markoviens. En revanche, si une transition immédiate est sensibilisée par un marquage le taux de transition associé vaut automatiquement l'infini. Le marquage correspondant est alors un marquage virtuel.

On constate qu'il n'existe pas de suite infinie de marquages virtuels successivement atteignables par des transitions immédiates, donc le processus de marquage des états tangibles est un processus de Markov homogène.

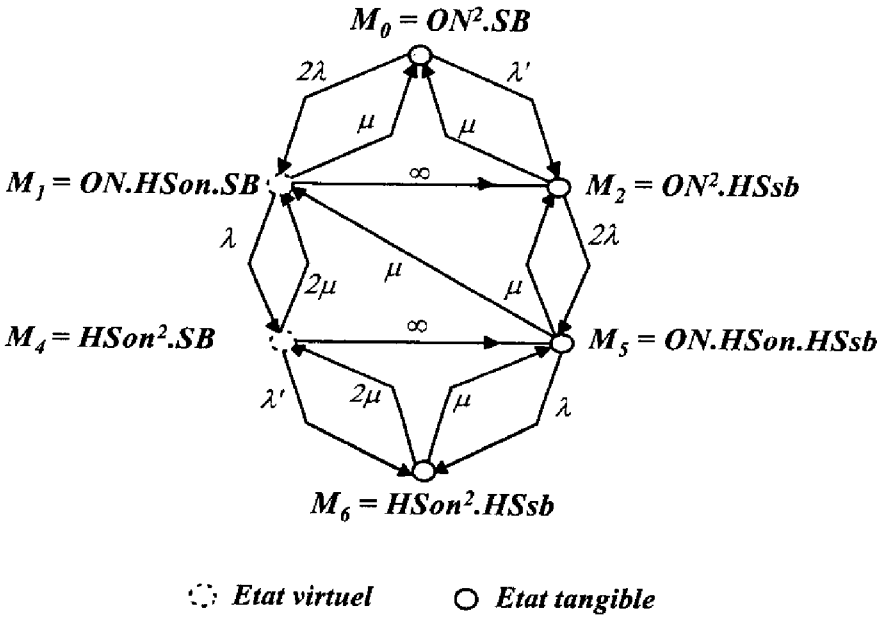


Figure IV.6: Graphe des Taux de Transition

Le graphe de Markov des états tangibles est obtenu après élimination des marquages virtuels. Nous illustrons ce principe d'élimination en nous basant sur un algorithme d'élimination portant sur le graphe des taux de transition déjà généré.

S'il n'y a qu'une transition immédiate t au départ d'un marquage virtuel M, tout autre transition partante est supprimée et le sommet de M est fusionné avec celui du marquage M' obtenu par le tir de t. Les arcs entrants de M sont conservés et ajoutés aux arcs entrants de M'. Si ces nouveaux arcs étaient déjà des arcs entrants de M' alors leur taux de transition sont sommés. Par exemple, le marquage M_1 *est un marquage virtuel* donc les arcs sortant vers M_0 et M_4 sont supprimés, le sommet de M_1 est fusionnés avec celui de M_2 (marquage obtenu par tir de la transition immédiate C) et le nouvel arc entrant de M_0 vers M_2 vaut maintenant $2\lambda+\lambda'$.

S'il y a plusieurs transitions immédiates au départ de M alors c'est qu'il existe un conflit entre transitions immédiates qui a pu être préalablement probabilisé. Dans ce cas le sommet de M est fusionné avec chacun des sommets des marquages obtenus par le tir de ces transitions. Ceci est réalisé comme précédemment, mais les taux de transitions entrant de chaque marquage M' provenant de ceux de M' sont pondérés par la distribution discrète associée à la transition immédiate menant de M à M'.

Cet algorithme d'élimination est mené jusqu'à ce que tous les marquages virtuels aient été supprimés. Le graphe de Markov des états tangibles obtenu pour notre exemple est donné par la Figure IV.7.

Un algorithme plus performant permettant de construire directement le graphe de Markov des états tangibles à partir du RdPSG sans générer le graphe des taux de transition est présenté dans [Ajmone 87]. Ceci, lorsque le nombre d'états virtuels est très important, permet de réduire considérablement la taille mémoire nécessaire.

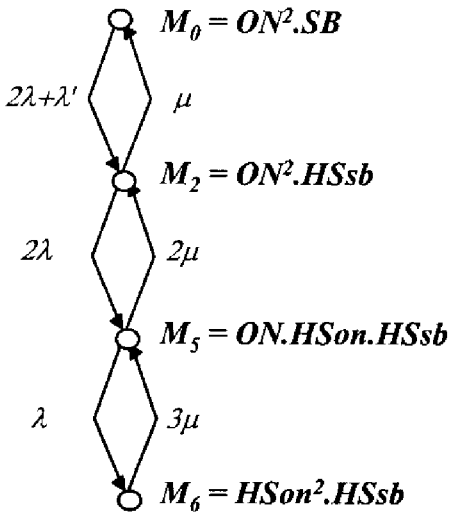


Figure IV.7: Elimination des Etats Virtuels - Graphe de Markov des Etats Tangibles

Ce graphe de Markov obtenu nous permet donc d'analyser le processus de marquage et en particulier de calculer la Disponibilité Instantanée de ce système par résolution de l'équation différentielle II.5.

Le générateur infinitésimal A s'obtient directement:

$$A = \begin{array}{c} \begin{matrix} M_0 & M_2 & M_5 & M_6 \end{matrix} \\ \begin{bmatrix} -(2\lambda+\lambda') & (2\lambda+\lambda') & 0 & 0 \\ \mu & -(\mu+2\lambda) & 2\lambda & 0 \\ 0 & 2\mu & -(2\mu+\lambda) & \lambda \\ 0 & 0 & 3\mu & -3\mu \end{bmatrix} \end{array} \begin{matrix} M_0 \\ M_2 \\ M_5 \\ M_6 \end{matrix}$$

L'équation II.5 nous donne la probabilité instantanée de chaque marquage :

$$\dot{P}(t) = \left[\dot{P}(M_0) \quad \dot{P}(M_2) \quad \dot{P}(M_5) \quad \dot{P}(M_6) \right] = P(t) \times A \qquad \textbf{(II.5)}$$

$$\text{avec} \quad P(0) = [1 \quad 0 \quad 0 \quad 0]$$

D'où, la Disponibilité $A(t)$ du système sachant que les états M_0 et M_2 sont les états de bon fonctionnement (2 éléments actifs):

$$A(t) = P(t) \times I_2 \qquad \text{avec} \quad I_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

La Figure suivante nous donne la Disponibilité Instantanée obtenue pour

$$\lambda = 1.15 \times 10^{-5}, \qquad \lambda' = \frac{\lambda}{10} \qquad \text{et} \qquad \mu = 6.85 \times 10^{-5}.$$

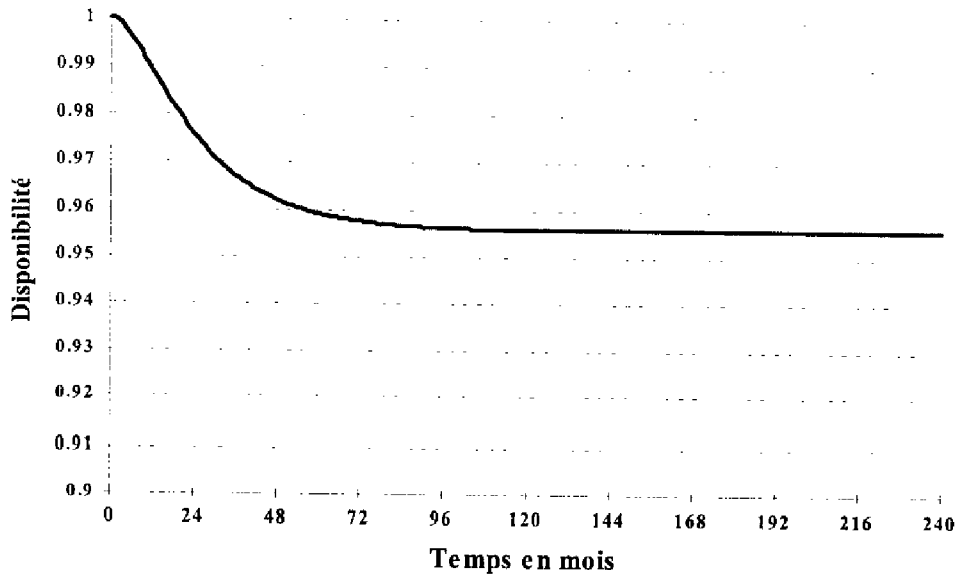


Figure IV.8 : Disponibilité Instantanée de la Redondance Passive *2* parmi *3*

On constate que, après un régime transitoire assez long (environ 4 ans), on obtient un régime stationnaire autour de *0,955*.

II.3.4. Conclusion

Les réseaux de Petri Stochastiques Généralisés ont connu, ces dernières années, un réel succès particulièrement dans la communauté des chercheurs. En effet, grâce à ce modèle, on peut bénéficier, d'une part, du pouvoir de modélisation et de validation offert par les réseaux de Petri (sur lesquels on a pu insister dans le chapitre précédent) et, d'autre part, de l'efficacité du calcul Markovien.

C'est ainsi que de nombreux outils, basés plus ou moins directement sur ce modèle, ont vu le jour dans les laboratoires de recherche. Parmi eux, on peut citer:

- SURF2 [SURF-2 93], basé sur le modèle RdPSG et les graphes de Markov.
- GreatSPN [Chiola 91], intégrant les récentes évolutions de la recherche.
- Ultra-SAN [UltraSan 95], basé sur un modèle dérivé des RdPSG.
- TOMSPIN [TOMSPIN 92], permettant de traiter jusqu'à plusieurs millions d'états.

Cependant, même si les outils académiques font légion, ils n'ont pas réellement percé dans le monde industriel. La principale raison tient aux limitations imposées par l'hypothèse Markovienne qui deviennent rapidement irréalistes dans la description d'un système réel.
Et si on veut s'affranchir de ces contraintes, le seul recours pour des systèmes de taille et de complexité importantes est, comme on l'a déjà vu, l'approche statistique.

II.4. Approche Statistique

II.4.1. Introduction

La justification et les principes de la simulation pour l'évaluation de systèmes stochastiques ont été présentés au chapitre II. Nous centrons donc ici notre propos sur cette approche statistique dans le cas particulier des réseaux de Petri.

On a vu que la mise en œuvre de la simulation se base sur l'exploitation d'un modèle logique qui décrit le comportement du système. Dans la démarche classique, on utilise généralement comme modèle un programme informatique spécialement écrit, car une des justifications majeures du recours à la simulation est liée aux limitations de modélisation et de prise en compte des hypothèses non Markoviennes des modèles classiques.

Les réseaux de Petri Stochastiques, comme on a pu le souligner au cours de ces deux derniers chapitres, conjuguent un fort pouvoir de modélisation à une prise en compte du temps très large. Ce modèle offre donc un bon support pour la simulation dont la mise en œuvre se base directement sur leur algorithme d'évolution. En effet, ce dernier décrit de façon non ambiguë les règles d'évolution d'un réseau bien au delà de l'hypothèse Markovienne, grâce aux règles d'évolution du modèle de base des réseaux de Petri et aux politiques d'exécution et de mémoire définies.

La simulation des réseaux de Petri Stochastiques, grâce aux larges possibilités du modèle, a d'avantage séduit les industriels. Des outils commerciaux se sont développés comme, en France, MOCA-RP de ELF ou, aux Etats-Unis, Design-CPN.

Nous présentons ici plus en détail le logiciel MISS-RdP qui est celui qui nous a servi durant ce travail et au développement duquel nous avons participé.

II.4.2. Le Simulateur MISS-RdP

MISS-RdP (Modélisation Interactive pour la Simulation de Systèmes par Réseaux de Petri) est développé, depuis 1991, par la société IXI sur la base d'un partenariat.
Ce partenariat regroupe aujourd'hui plusieurs industriels du spatial (ALCATEL ESPACE, le CNES et AEROSPATIALE), le centre de la navigation aérienne (CENA) ainsi que les laboratoires du LAAS-CNRS et du GAPSE de l'ENSEEIHT qui apportent leur expertise technique.
La dernière version [IXI 95] supporte la simulation d'un modèle dérivé des réseaux de Petri à Objets [Sibertin 85]. Nous nous limiterons ici à la description de la version de base.
MISS-RdP est principalement utilisé par les partenaires pour les études de Sûreté de Fonctionnement (et notamment de Disponibilité) de systèmes non Markoviens et fortement distribués.

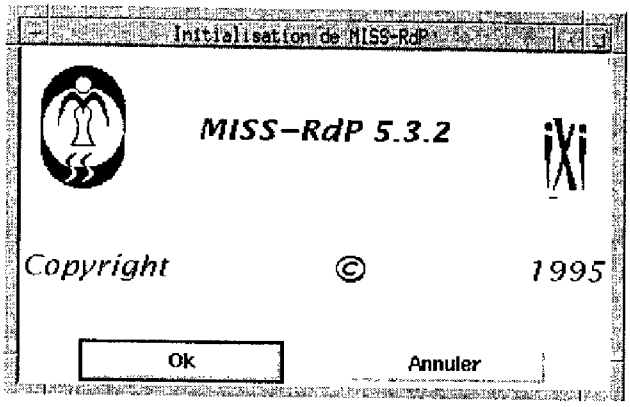


Figure IV.9 : Invite du Logiciel MISS-RdP

II.4.2.1 Algorithme d'Evolution

Le fonctionnement de MISS-RdP est directement basé sur l'algorithme d'évolution présenté en IV.2.2. C'est à dire qu'il est basé sur le modèle RPTS présenté au paragraphe IV-2. Il suit donc une politique d'évolution basée sur le *modèle concurrentiel* et une politique de mémoire d'*Enabling Memory*.
La réalisation des variables aléatoires est effectuée grâce à un générateur de nombres aléatoires dont les principes ont été décrits en II.2.3.4. Les dates de tir de transition ainsi générées sont gérées à partir d'un échéancier global.

II.4.2.2 Transitions Multi-Sensibilisées

La prise en compte des transitions multi-sensibilisées n'est pas possible sur la version de base: une seule date de tir peut être associée à chaque transition.
Grâce à la version des réseaux à Objets où chaque jeton est différencié, une date de tir est maintenant associée à chaque n-uplet de jeton permettant ainsi de prendre en compte la multi-sensibilisation.

II.4.2.3 Transitions en conflit

Lorsque plusieurs transitions (ou couple n-uplet/transition) sont franchissables au même instant, par défaut, on choisit la prochaine transition à franchir selon un tir aléatoire uniforme. C'est le cas notamment si ces transitions sont en conflit.
Il est toutefois possible de probabiliser, à priori, un conflit potentiel en utilisant une transition indéterministe décrite par la figure suivante.

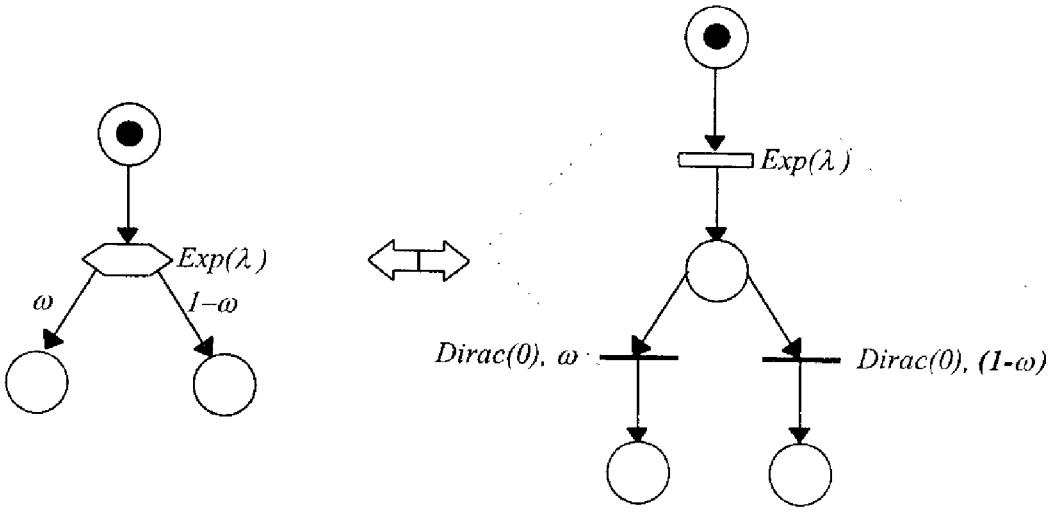


Figure IV.10 : Exemple de Transition Indéterministe et Représentation Classique Associée

Une transition indéterministe est donc une « macro » qui permet de probabiliser un conflit entre plusieurs transitions ayant le même ensemble de places amonts. Ceci est réalisé en associant aux arcs avals d'une telle transition une distribution discrète de probabilité.

II.4.2.4 Grandeurs Observées

MISS-RdP permet deux types d'observation de l'évolution du réseau.

- L'observation des états. Un état est une variable booléenne décrivant une partie de l'espace d'état du processus de marquage. Il est défini par l'utilisateur grâce à une formule arithmétique et logique portant sur le marquage des places du réseau. Exemple : $E_1 = M(ON) \geq 2$, cet état permettrait d'évaluer la Disponibilité du système décrit par la Figure IV.4.
- L'observation des événements. Un événement est une variable entière définie à partir d'une liste de transitions. Cette variable est incrémentée à chaque franchissement de l'une des transitions de la liste. Exemple: $Ev_1 = \{D_{on}, D_{sb}\}$, cet événement permettrait d'obtenir une statistique sur le nombre de défaillances survenues sur le système décrit par la Figure IV.4.

II.4.2.5 Estimateurs et Précisions des Résultats

L'utilisateur, après avoir défini les états, les événement ainsi que les paramètres de simulations (nombre d'histoires N et horizon de simulation T cf. II.2.3.1), saisit les dates θ auxquelles il veut observer son réseau. MISS-RdP permet alors d'estimer à ces dates certaines quantités associées aux états et événements. Nous présentons ici, celles couramment utilisées.

Soit E un état, Ev un événement et θ une date de calcul, le logiciel évalue:

- La valeur moyenne :

$$\hat{\Psi}_E(\theta) = \frac{1}{N} \cdot \sum_{k=1}^{N} E_k$$

avec E_k l'état E mesuré à la date θ pour l'histoire numéro k.
Pour l'état E_1 défini ci-avant, cet estimateur nous donnerait la Disponibilité du système à la date θ.

- La valeur moyenne :

$$\hat{\zeta}_E(\theta) = \frac{1}{N} \cdot \sum_{k=1}^{N} W_k$$

avec W_k variable booléenne testant si l'état E est continuellement resté à vrai sur l'intervalle $[0,\theta]$.
Pour l'état E_1 défini ci-avant, cet estimateur nous donnerait la Fiabilité du système à la date θ.

- La valeur moyenne et l'écart type:

$$\hat{m}_{Ev}(\theta) = \frac{1}{N} \cdot \sum_{k=1}^{N} Ev_k \text{ et } \hat{\sigma}_{Ev}(\theta) = \frac{1}{N-1} \cdot \sum_{k=1}^{N} \left[Ev_k^{\,2} - \hat{m}_{Ev}^{\,2}(\theta) \right]$$

avec Ev_k l'événement mesuré à la date θ pour l'histoire numéro k.
Pour l'événement Ev_1 défini ci-avant, cet estimateur nous donnerait le nombre moyen de défaillances à la date θ.

La précision de ces estimateurs, qui est de première importance, a fait l'objet d'une étude financée par les partenaires MISS-RdP [Moyse 95]. Nous rappelons ici l'expression des intervalles de confiance obtenue pour chacun des estimateurs présentés.

Les valeurs u_α utilisées dans les formules suivantes sont définies à partir des tables de la loi normale par ($(1-\alpha)$ étant le niveau de l'intervalle de confiance considéré) :

$$F(u_\alpha) = 1 - \frac{\alpha}{2}$$

- $\hat{\Psi}_E(\theta)$:

Pour tout état E, pour toute date θ, pour N grand et si $N.\hat{\Psi}_E(\theta) > 5$ et $N\left(1 - \hat{\Psi}_E(\theta)\right) > 5$, l'intervalle de confiance au niveau $(1-\alpha)$ est donné par:

$$\left[\hat{\Psi}_E(\theta) - u_\alpha \sqrt{\frac{\hat{\Psi}_E(\theta).\left(1 - \hat{\Psi}_E(\theta)\right)}{N}},\ \hat{\Psi}_E(\theta) + u_\alpha \sqrt{\frac{\hat{\Psi}_E(\theta).\left(1 - \hat{\Psi}_E(\theta)\right)}{N}} \right] \qquad \textbf{(IV.1)}$$

Si N est grand, mais sans vérifier les hypothèses précédentes, on a l'intervalle de confiance au niveau $(1-\alpha)$:

$$\left[\hat{\Psi}_E(\theta) - \frac{1}{\sqrt{\alpha}}\sqrt{\frac{\hat{\Psi}_E(\theta).\left(1-\hat{\Psi}_E(\theta)\right)}{N}},\ \hat{\Psi}_E(\theta) + \frac{1}{\sqrt{\alpha}}\sqrt{\frac{\hat{\Psi}_E(\theta).\left(1-\hat{\Psi}_E(\theta)\right)}{N}}\right]$$

(IV.2)

- $\hat{\zeta}_E(\theta)$:

Pour cet estimateur, les formules (IV.1) et (IV.2) s'appliquent également en remplaçant $\hat{\Psi}_E(\theta)$ par $\hat{\zeta}_E(\theta)$.

- $\hat{m}_{Ev}(\theta)$:

Pour tout état E, pour toute date θ, pour N grand l'intervalle de confiance au niveau $(1-\alpha)$ est donné par:

$$\left[\hat{m}_{Ev}(\theta) - \frac{1}{\sqrt{\alpha}}.\frac{\hat{\sigma}(\theta)}{\sqrt{N}},\ \hat{m}_{Ev}(\theta) + \frac{1}{\sqrt{\alpha}}.\frac{\hat{\sigma}(\theta)}{\sqrt{N}}\right] \qquad \textbf{(IV.3)}$$

II.4.3. Exemples

II.4.3.1 Précisions des Résultats

Afin de proposer une comparaison entre traitement analytique et une approche par simulation et d'illustrer la précision des résultats, on reprend ici notre exemple du §IV.3.3.

Les simulations ont été réalisées avec les taux de défaillance et de réparation précédemment définis et pour les paramètres de simulation suivants:

- un horizon de simulation T de *20* ans,
- un pas d'observation de *1* mois,
- et pour $N = 10000$ histoires.

La Figure IV.11 compare les courbes de Disponibilité obtenues par calcul Markovien et par simulation. Les courbes « + » et « - » définissent l'intervalle de confiance à 95% calculé grâce à la formule IV.1.
L'état observé correspond ici au test de marquage: $E_1 = M(ON) \geq 2$.

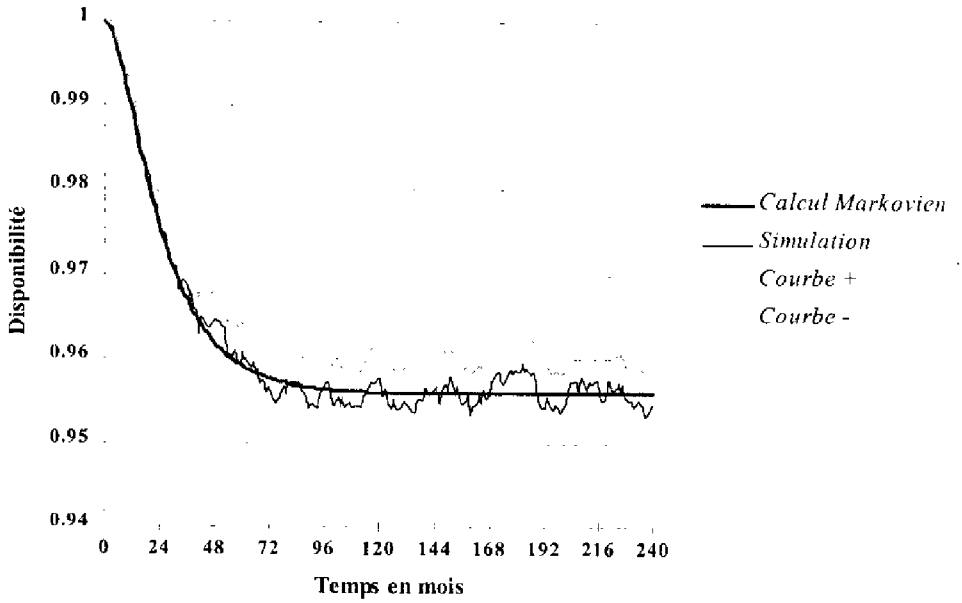


Figure IV.11 : Précision sur la Disponibilité

Comme le réseau simulé est Markovien, on peut ici vérifier la bonne qualité des résultats de simulation par rapport au calcul exact. Notons toutefois que cette simulation ne se justifie qu'afin d'illustrer notre propos. En effet, lorsque l'approche analytique est possible elle sera toujours préférable à la simulation qui ne fournit que des résultats estimés pour des temps de traitement nettement supérieur aux temps de calcul.

La Figure IV.12 donne une estimation par simulation du nombre moyen de remplacement d'éléments nominaux devenus défaillants ainsi que l'intervalle de confiance à 95% associé.
L'événement défini correspond ici à la variable : *nombre de tirs de la transition Ron.*

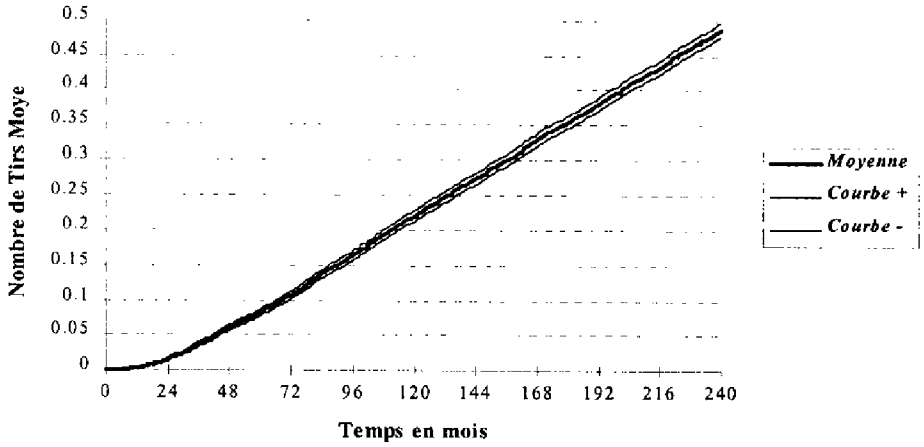


Figure IV.12 : Précision sur le Nombre de Tirs de Transition

II.4.3.2 Réseau non-Markovien

Le principal intérêt de la simulation est de proposer un recours lorsque l'approche analytique n'est plus possible. Nous présentons un exemple de réseau non Markovien pour lequel la simulation s'avère tout à fait pertinente.

Ce réseau est décrit par la Figure IV.13. Il s'agit toujours du réseau de la redondance passive *2* parmi *3*, mais on suppose maintenant que les éléments nominaux ont, en plus de leur taux de défaillance, une durée de vie maximale fixée. De plus, la commutation décrite par la transition *C*, au lieu d'être immédiate s'effectue au bout d'une durée fixe *c*. Enfin, les réparations interviennent après une durée fixe correspondant à la valeur moyenne de la loi exponentielle précédemment utilisée. Ces modifications sont prises en compte sur le nouveau réseau grâce aux éléments suivants:

- Une transition *Fdv* modélisant la durée de vie maximale a été rajoutée entre la place *ON* et *HSon*. Une loi de Dirac de paramètre *f* est associée à cette dernière (on fixe $f = 61320h \approx 7\ ans$).
- Les lois associées aux transitions décrivant la réparation d'éléments (*Ron* et *Rsb*) sont maintenant des lois de Dirac de paramètre $1/\mu$ (on fixe $1/\mu = 14600h \approx 1{,}5\ an$).
- La transition de commutation *C* n'est plus une transition immédiate, mais une transition de Dirac de paramètre *c* (on fixe $c = 504h \approx 3\ semaines$).

Les modifications effectuées ne portent que sur les spécifications temporelles. Le comportement logique du réseau reste inchangé et le réseau étendu sous-jacent est identique dans les deux cas.

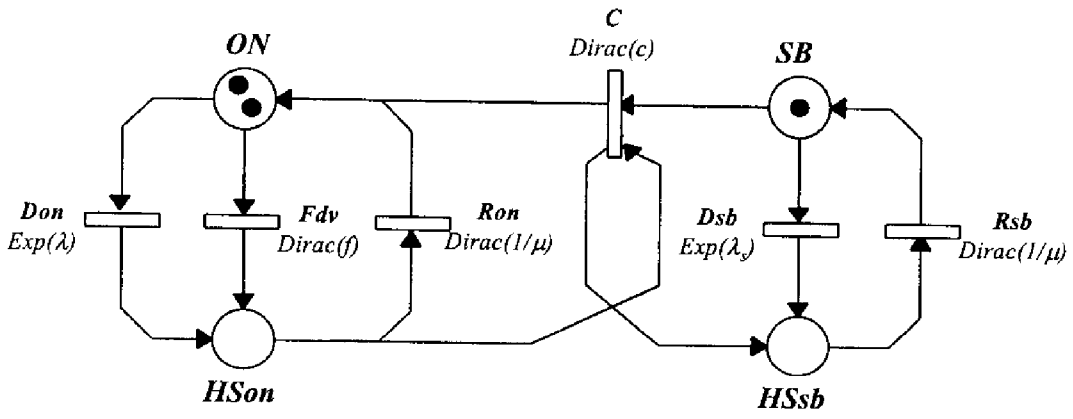


Figure IV.13 : Exemple de Réseau non Markovien

La Figure IV.14 compare les résultats de Disponibilité obtenus avec le cas Markovien précédemment étudié et la simulation du réseau ci-dessus.

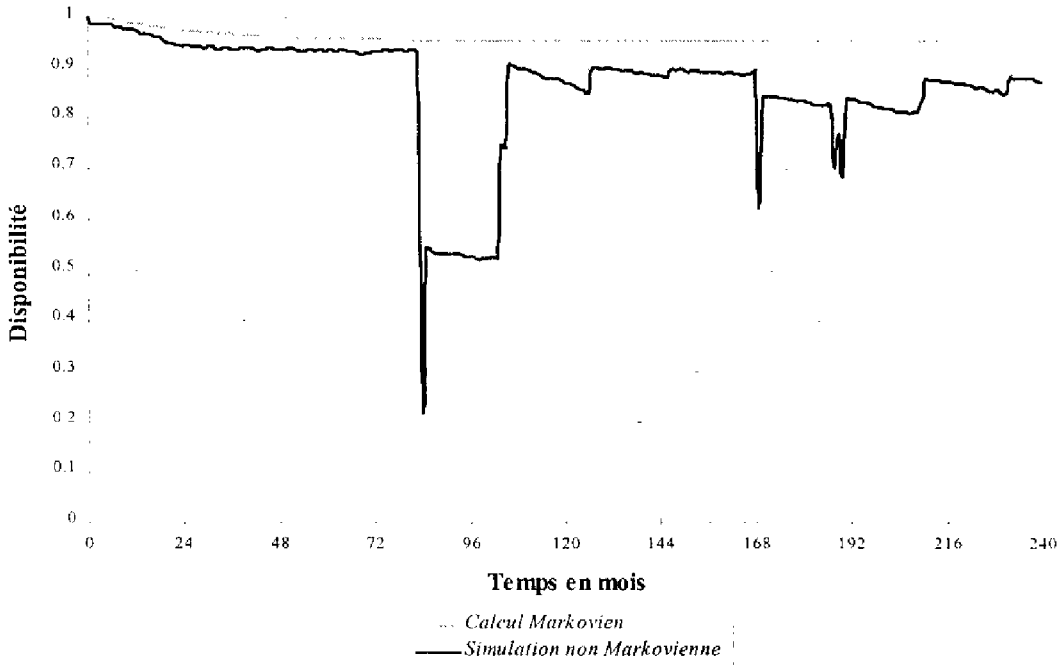


Figure IV.14 : Comparaison de Disponibilité entre le cas Markovien et non Markovien

On constate que ces deux courbes sont réellement dissemblables.
La courbe obtenue par calcul Markovien ne prend pas en compte la fin de vie des éléments nominaux ainsi que la durée de commutation. La seule façon, avec les réseaux de Petri Stochastiques Généralisés, d'intégrer ces contraintes temporelles eût été de représenter ces durées par des transitions temporisées (loi exponentielle associée). Ainsi, on pourrait représenter la fin de vie en ajoutant $1/f$ au taux de défaillance λ de la transition *Don*. La courbe alors obtenue aurait une allure similaire à celle présentée ci-dessus, légèrement décalée vers le bas.
Or, l'allure de la courbe de Disponibilité obtenue par simulation est singulièrement différente. Ceci est en particulier du à la fin de vie des éléments nominaux qui est une contrainte déterministe. C'est pourquoi on constate les creux sur la courbe qui débutent à $\theta = 84$ *mois* (=*7ans*). Au total la Disponibilité résultante est nettement en dessous de celle obtenue par calcul Markovien avec des chutes qui peuvent être tout à fait critiques pour le système et qui peuvent justifier la mise en œuvre d'actions préventives importantes.

Donc, sur cet exemple, l'approche analytique n'aurait pas permis de mettre en évidence ces singularités. Dans certains cas, la simulation permet bien plus que d'affiner des résultats obtenus par approximations analytiques elle remet complètement en question ces derniers. Elle se justifie alors dès les premières modélisations. Nous reviendrons sur ce point dans le chapitre qui suit.

II.1. Conclusions

Nous avons présenté au cours de ce chapitre les réseaux de Petri Stochastiques qui sont une extension temporelle et probabiliste du modèle logique des réseaux de Petri Etendus exposés dans le chapitre précédent.

Moyennant certaines hypothèses probabilistes, à partir desquelles sont définis les réseaux de Petri Stochastiques Généralisés, ce modèle, tout en permettant de profiter du pouvoir de modélisation des réseaux de Petri, reste entièrement compatible avec l'approche Markovienne présentée au cours du chapitre II. Ainsi, un système distribué Markovien présentant une forte explosion combinatoire du nombre des états pourra être décrit plus aisément par réseau de Petri et directement évalué sans avoir à énumérer à la main l'ensemble de ses états. La seule limite est celle imposée par le calculateur en charge du traitement.

A partir de règles d'évolution clairement définies (cf. algorithme d'évolution §IV.2.2) le modèle général des Réseaux de Petri Stochastiques constitue un bon support pour la simulation. En effet, un simulateur de réseaux (comme MISS-RdP) peut être écrit une fois pour toutes et l'approche statistique devient un processus relativement rapide pour lequel il n'est plus besoin de concevoir un programme dédié mais simplement de se concentrer sur la conception du modèle.

Parés ces présentations de concepts théoriques, nous allons maintenant illustrer de façon plus concrète les possibilités des réseaux de Petri Stochastiques à partir d'une étude de cas propre à notre domaine d'application.

Chapitre V

Disponibilité Opérationnelle d'une Constellation de Satellites

V.1. Introduction

Emergence des Systèmes à base de Constellation de Satellites

Depuis quelques années, de nombreux projets spatiaux (plus d'une quarantaine) basés sur des constellations de satellites sont à l'étude, en cours de réalisation ou déjà en service. Les progrès techniques considérables de l'industrie spatiale réalisés sur les coûts, la taille, la masse et les performances des satellites permettent en effet d'envisager de tels systèmes. Grâce à une très grande couverture (qui peut être mondiale) continue ou quasiment continue, ils permettent de démultiplier les possibilités des différents types de missions spatiales.

Ceci est particulièrement vrai dans le domaine des télécommunications où les constellations de satellites en orbite basse viennent compléter avantageusement les réseaux terrestres et les systèmes géostationnaires. En effet, elles permettent d'offrir les différents services (téléphonie, localisation, messagerie, télédiffusion, multimédia, ...) dans les zones disposant de peu d'équipements terrestres comme dans les pays en voie de développement ou les zones à faible densité de population. De plus, ces services sont directement accessibles aux utilisateurs grâce à des terminaux peu encombrants et de puissance réduite en raison des faibles distances terminal - satellite (entre 500 et 2000 km par rapport à 36000 km pour les satellites géostationnaires).

Ainsi, avec la croissance très forte de la demande et la libéralisation mondiale des télécommunications, des marchés potentiels très importants se profilent pour ces systèmes. Forts de ce constat, de nombreux projets se dessinent, laissant présager une concurrence ardue dans les années à venir. Parmi eux, on peut citer GLOBALSTAR et IRRIDIUM pour la

téléphonie, TELEDESIC et SATIVoD pour le multimédia ou encore ORBCOMM et STARSYS pour la localisation et la messagerie.

Disponibilité Opérationnelle

Si ces systèmes sont très prometteurs, ils doivent néanmoins relever des défis techniques et technologiques très importants pour pouvoir offrir une qualité de service qui permette de séduire les utilisateurs potentiels. Cette qualité de service, plus généralement appelée disponibilité de service [Dosière 95], dépend de nombreux facteurs comme la visibilité géométrique de la constellation, les contraintes radioélectriques de propagation, les protocoles de communication ou encore la capacité du système en terme de trafic. Parmi ces facteurs, le bon fonctionnement des différents segments du système, c'est à dire sa Disponibilité Opérationnelle, joue un rôle bien évidemment de premier plan. Et le segment spatial, en raison des contraintes déjà évoquées (cf. chapitre I), est celui dont le bon fonctionnement est le plus difficile à assurer.

Or, l'étude de la Disponibilité Opérationnelle du segment spatial d'un système à base de constellation de satellites dépasse celle de chaque satellite pris isolément. En effet, il convient maintenant de décrire, en plus des possibilités de défaillances du véhicule spatial, les stratégies et processus de mise en place et de renouvellement des satellites suite à défaillance ou à fin de vie. C'est face à une telle problématique et à l'impossibilité d'utiliser les méthodes classiques d'études de Disponibilité, que le CNES et ALCATEL ESPACE se sont intéressés aux réseaux de Petri, chacun d'eux étant en effet impliqué dans l'étude voire le développement de systèmes spatiaux à base de constellation de satellites. C'est ainsi que nous avons été amenés à effectuer ce type d'étude par réseaux de Petri sur les systèmes suivants:

- TPFO, BIMILSAT [Ereau 93 (2)],
- GLOBALSTAR [Ereau 94 (2)],
- STARSYS [Ereau 93] [Ereau 95 (3)].

Objectif et Organisation du chapitre

Au cours des deux chapitres précédents nous avons présenté les réseaux de Petri à partir d'un exemple « académique » qui a permis de mettre en relief différents concepts théoriques. C'est maintenant pour nous l'occasion d'illustrer de façon plus pragmatique leur utilisation sur un type d'application très concret, de complexité significative et de première actualité dans l'industrie spatiale [Ereau 96]. On va pouvoir notamment aborder le problème de la construction de réseaux de taille significative.

Nous décrivons tout d'abord le système considéré en soulignant les caractéristiques intervenant dans le bon fonctionnement du segment spatial. Ce système est volontairement très générique, ceci afin, d'une part, de nous centrer sur la problématique évoquée et, d'autre part, de souligner la souplesse de modélisation par réseaux de Petri.

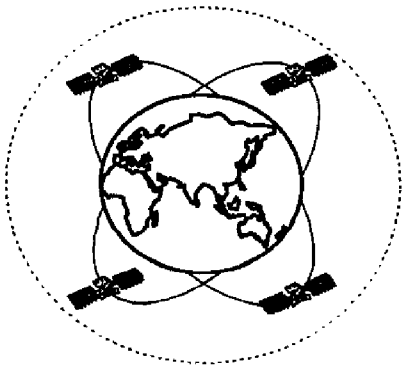
Les deux sections suivantes sont consacrées à la construction des modèles. Dans un premier temps on décrit un scénario simplifié de déploiement et de maintenance de la constellation. Ceci nous permet de construire et de vérifier de façon simple et progressive un premier modèle. Sur la base de cette première étape, le scénario est ensuite enrichi afin de décrire plus finement le système. Le modèle global est revu et augmenté pour s'adapter à ces nouvelles contraintes. La complexité qu'il présente alors limite les possibilités de vérification formelle et pourtant, elle est représentative de celle obtenue sur des modèles de projets réels.

La dernière section est consacrée à l'évaluation des modèles construits. Après avoir justifié le recours à la simulation, on illustre le type de résultats qui nous est accessible et leur intérêt dans un contexte d'aide au dimensionnement.

V.2. Description du Système

La Figure V.1 décrit les composantes intervenant dans le bon fonctionnement du segment spatial de la constellation étudiée ici. On peut les regrouper en deux grands ensembles:

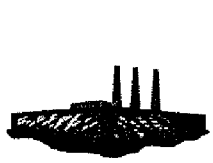
- *Le Segment Spatial* lui-même et
- *Le Soutien Logistique Sol.*



SEGMENT SPATIAL :
Constellation de *p* plans d'orbite avec *k* parmi *n* satellites chacun

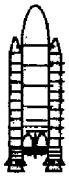
Demande de Satellites

Offre de Satellites



***c* chaînes de Production**

possibilités de stock



***l* aires de lancement**

SOUTIEN LOGISTIQUE SOL : ressources sol

Figure V.1: Segment Spatial et le Soutien Logistique Sol

Le *Segment Spatial* regroupe les satellites nécessaires à la mission ainsi que les satellites de secours. Il est constitué de p plans d'orbite pouvant contenir chacun n satellites. Parmi ces n satellites k sont actifs et $(n-k)$ sont en attente prêts à prendre le relais en cas de défaillance. La défaillance d'un satellite est caractérisée par un taux λ pour un satellite actif et un taux λ' pour un satellite en attente. Lors de la défaillance d'un satellite actif, un satellite en attente sur le même plan d'orbite peut remplacer ce dernier après une durée de test et de manoeuvre notée T_{change}. La durée de vie (hors défaillances) des satellites est conditionnée par leur consommation en ergol utilisé pour les manoeuvres de correction d'orbite. Ces manoeuvres sont nécessaires pour les satellites actifs mais pas pour les satellites en attente: on considère donc que seuls les satellites actifs ont une durée de vie limitée.

Le *Soutien Logistique Sol* a pour rôle la production, le stockage et la mise en orbite des satellites. Pour cela il dispose d'un nombre de ressources limité: c chaînes de production ainsi que l aires de lancement. On note T_{prod} la durée nécessaire pour produire un lot de k satellites,

T_{comm} celle pour disposer d'un lanceur de k satellites, T_{camp} la durée de campagne lanceur et enfin p_{launch} la probabilité de succès au lancement.

Cette description fait apparaître deux types de paramètres:

- Les paramètres discrets qui interviennent dans la modélisation du système : à savoir $\{p, n, k\}$ associés au *Segment Spatial* et $\{c, l\}$ associés au *Soutien Logistique Sol*,
- Les paramètres Temporels et Stochastiques qui interviennent dans l'évaluation du système. Ils sont rappelés dans le tableau qui suit.

Paramètre	***Commentaire***
SEGMENT SPATIAL	
λ	Taux de défaillance satellite nominal
λ'	Taux de défaillance satellite de secours
T_{life}	Durée de vie d'un satellite nominal
T_{change}	Durée pour activer un satellite en attente
SEGMENT SOUTIEN LOGISTIQUE	
T_{prod}	Durée pour produire k satellites
T_{comm}	Délai pour avoir un lanceur disponible
T_{camp}	Durée de la campagne de lancement
p_{launch}	Probabilité de succès du lanceur

Tableau V.1: Paramètres Temporels et Stochastiques

Nous venons de décrire les composantes, ainsi que les données temporelles et stochastiques associées, intervenant dans le bon fonctionnement du segment spatial. Un nombre infini de stratégies utilisant les ressources énoncées peuvent être imaginées pour pallier les défaillances des satellites et des lanceurs. Dans ce qui suit, nous proposons un certain type de scénario qui n'est vraisemblablement pas optimal mais qui permet d'illustrer simplement des stratégies assez complexes d'utilisation des ressources. Il suppose que la production des satellites et la commande des lanceurs ne sont pas planifiées au préalable et se basent directement sur les besoins de la constellation. Ceci, dans un contexte industriel, n'est bien sûr pas envisageable mais peut permettre d'aider à définir le nombre de satellites et de lanceurs nécessaires à la maintenance de la constellation et ainsi aider à la planification de ces besoins supplémentaires (cf. Figure V.2).

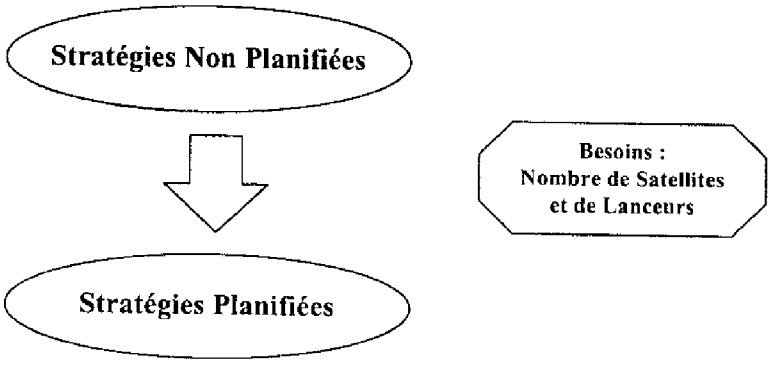


Figure V.2: Evolution des Scénarios dans ce Type d'Etude

V.3. Scénario Simplifié

V.3.1. Description

On présente ici un scénario très simple de déploiement et de renouvellement de la constellation. Il n'est pas imaginable dans un cas réel mais permet de poser progressivement le problème.

Déploiement de la Constellation

On suppose qu'initialement la constellation n'est pas déployée, aucun satellite ni lanceur n'a été stocké, produit ou même commandé. Le processus de déploiement consiste à positionner k satellites actifs sur chacun des p plans d'orbite.
Les satellites sont produits et lancés par lot de k jusqu'à ce que chaque plan ait été rempli.
En cas de défaillance d'un lanceur, k nouveaux satellites sont produits et lancés pour pallier cet échec. Ceci induit donc un retard au déploiement.
On suppose qu'un lanceur est toujours disponible.
On ne fait pas de stockage de satellites de secours au sol.

Remplacement suite à défaillance

D'après le principe retenu pour le déploiement, lorsqu'une première défaillance survient sur l'un des plans d'orbite, aucun satellite de secours n'est présent sur ce même plan. On lance alors une demande de production et de tir de k nouveaux satellites. Lorsque ces k satellites arrivent sur le plan, l'un d'entre eux est activé et les $(k-1)$ autres sont mis en attente. A la prochaine défaillance d'un satellite actif sur ce plan, c'est l'un des satellites de secours qui va être utilisé. Quand il n'y a plus de satellites de secours (tous utilisés ou défaillants) une nouvelle demande de k satellites est effectuée.

Remplacement suite à Fin de Vie

On ne considère pas pour l'instant la durée de vie limitée des satellites actifs.

Le scénario présenté ici suppose que tous les tirs de satellites se font par lots de k. D'après ce qui a été décrit, cela impose que chaque plan d'orbite a au plus $(k-1)$ satellites de secours, ce qui fixe la valeur du nombre maximum de satellites sur un plan à $n=(2k-1)$.

V.3.2. Modélisation

Après cette description aride du fonctionnement du système (mais qui correspond aux descriptions textuelles que l'on rencontre dans tout projet), nous allons construire et vérifier par étapes un modèle possible pour ce scénario. C'est d'ailleurs une façon naturelle et efficace d'aborder tout problème d'une certaine complexité. On va ainsi pouvoir illustrer deux types de construction modulaire de réseaux de Petri: l'approche « bottom-up » par composition de réseaux et l'approche « top-down » par affinement de réseau.

Modèle d'un Plan d'Orbite

Comme point de départ de cette modélisation on se penche sur le comportement possible des satellites de chacun des p plans d'orbite. Tout plan peut comporter jusqu'à n satellites dont k peuvent être actifs en même temps. De plus, ces satellites peuvent défaillir et être remplacés soit par de nouveaux satellites, soit par des satellites de secours. Ce comportement est exactement celui décrit à plusieurs reprises au cours des deux derniers chapitres: on retrouve le principe de la redondance de k satellites parmi n rappelé par le réseau de la Figure V.3. Le marquage initial indique qu'aucun satellite n'a été déployé c'est à dire qu'aucun satellite n'est apte à fonctionner.

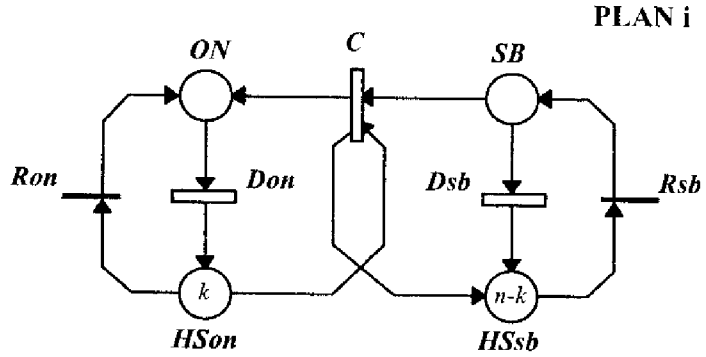


Figure V.3: Modèle d'un Plan d'Orbite

L'analyse qualitative de ce modèle menée au chapitre III nous a permis de prouver certaines propriétés qui correspondent au comportement souhaité. Rappelons les:

- jamais plus de k satellites peuvent être actifs en même temps, de même que jamais plus de $(n\text{-}k)$ satellites ne peuvent être en attente en même temps (invariants de places).
- Toutes les défaillances, commutations, remplacements restent possibles, il n'y a aucune dégradation irréversible (vivacité).

Ce modèle a maintenant besoin d'être enrichi afin de décrire plus précisément le scénario présenté.

Modèle d'un Plan d'Orbite et de ses Interfaces

Si l'on reste au niveau d'un plan, il est dit que:

1. Seulement les k satellites nominaux sont déployés au début,
2. Tout remplacement s'effectue par lot de k satellites.

Le réseau de la Figure V.4 modélise ce comportement tout en conservant celui décrit ci-dessus. Trois places et deux transitions ont été rajoutées au modèle précédent. La place *OUT*, complémentaire des places *ON* et *SB*, totalise le nombre de défaillances survenues à la suite dans le plan. Lorsque ce nombre est égal à k, une demande de satellites (matérialisée par le tir de la transition *Dem*) est réalisée. Ce sous réseau représente l'interface de sortie du plan.

La demande est mémorisée dans une place *MEM*. Lorsqu'elle est comblée (tir de la transition *Rec*) *k* jetons sont générés dans la place d'interface d'entrée. Ces jetons seront prélevés par les transitions *Ron* et *Rsb* (prioritairement par *Ron*) modélisant la mise à disposition de nouveaux satellites actifs et en attente.

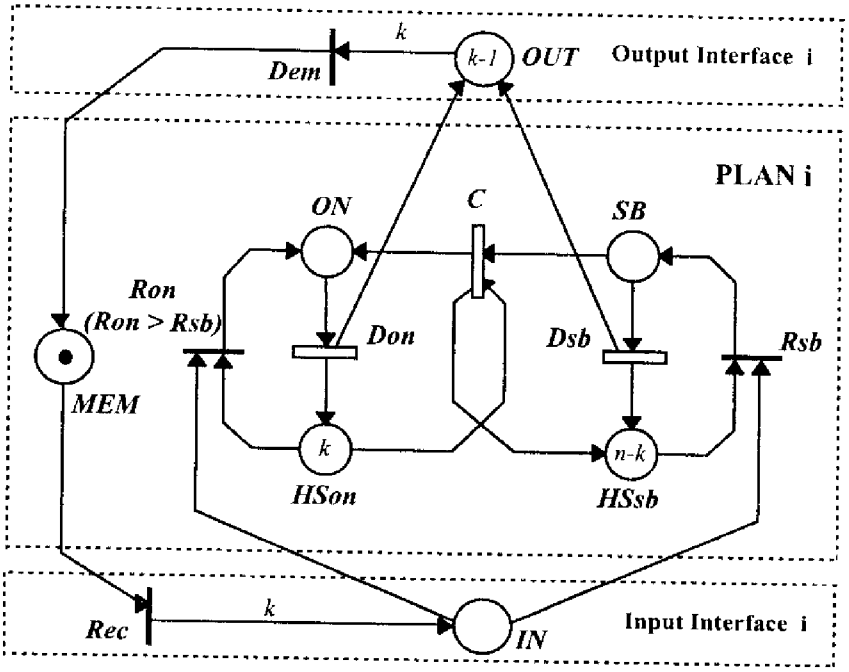


Figure V.4: Modèle d'un Plan et de ses Interfaces

Le marquage initial indique toujours qu'aucun satellite n'a été déployé dans le plan, mais de plus que *k* satellites ont été demandés (marquage de la place *MEM*). Lorsque ces *k* satellites seront disponibles, ils seront directement activés (tir de *Ron* et marquage à *k* de la place *ON*). Si une défaillance survient, comme la place de l'interface de sortie est marquée à *(k-1)*, une nouvelle demande de *k* satellites est réalisée.

Cette extension du réseau de la redondance ne modifie pas son comportement mais simplement précise la façon dont sont demandés et mis à disposition les nouveaux satellites. On peut montrer, par réduction de la séquence $OUT \xrightarrow{Dem} MEM \xrightarrow{Rec} IN$, que la place résultante « *OUT/IN* », qui contient alors *[(k-1)+k]* jetons, peut être supprimée. On obtient donc exactement le même réseau que celui de la Figure V.3.

Modèle du Segment Spatial

Le segment spatial étant composé de *p* plans, autant de ces modèles sont nécessaires pour décrire chacun d'entre eux c'est à dire que *p* réseaux comme celui de la Figure V.4 sont nécessaires pour décrire le comportement en orbite de tous les satellites. Toutefois ces réseaux ne peuvent être indépendants puisque chaque plan partage la même ressource *Soutien Logistique Sol* chargée du déploiement et du renouvellement des satellites. Il est donc nécessaire de les composer sans pour autant remettre en question les propriétés rappelées plus haut.

Le réseau de la Figure V.5 décrit cette composition. Deux réseaux représentant un plan *i* et un plan *j* ainsi que leurs interfaces ont été composés. Cette composition a été réalisée par fusion de places identiques aux places *MEM* de chacun des modèles. La place obtenue nommée *SLS* contient la somme des jetons des places *MEM*, donc deux.
On montre que ce type de fusion de place ne modifie en rien le comportement de chacun des deux réseaux.

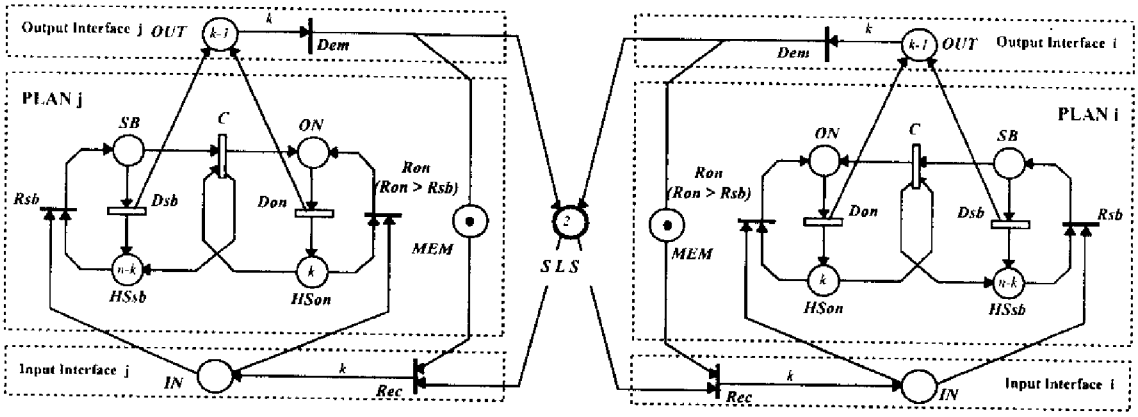


Figure V.5: Modèle du Segment Spatial

Un tel type de fusion peut être généralisé aux *p* réseaux décrivant chacun des plans de la constellation. La place *SLS* contient alors *p* jetons et représente la ressource partagée par chacun des plans et nécessaire à l'obtention de nouveaux satellites à savoir: le *Soutien Logistique Sol.*

Modèle du Soutien Logistique Sol

Le *Soutien Logistique Sol* est chargé, comme on l'a vu, de mettre à disposition pour le segment spatial, des lots de *k* satellites. Pour cela il doit les produire, les intégrer au lanceur (campagne), puis les lancer. En cas d'échec au lancement, k nouveaux satellites sont produits puis mis en orbite.

Le réseau de la Figure V.6 modélise ce comportement. Le modèle *Soutien Logistique Sol* a été obtenu par affinement de la place *SLS*. On retrouve la séquence *Production/Campagne/Tir*. Le tir est modélisé par un conflit entre deux transitions qui sera résolu de façon aléatoire après avoir probabilisé chacune de ces transitions (cf. § V.4.). Chacune des places de cette séquence est couverte par un invariant indiquant que jamais plus de *p* demandes ne peuvent être traitées en même temps.
Des règles de construction de réseaux par affinements successifs ont été définies par Valette [Valette 76] et développées par la suite [Suzuki 82], [Zhou 92]. Elles sont duales des règles de réduction de Berthelot et garantissent, sous certaines conditions, la conformité du comportement entre le réseau initial et le réseau dérivé. En les appliquant ici, on montre qu'aucun blocage n'a été introduit en dérivant ainsi la place *SLS*.
Ces règles peuvent également être appliquées sur les transitions *PRODUCTION* et *CAMPAGNE* pour décrire plus précisément chacun de ces processus.

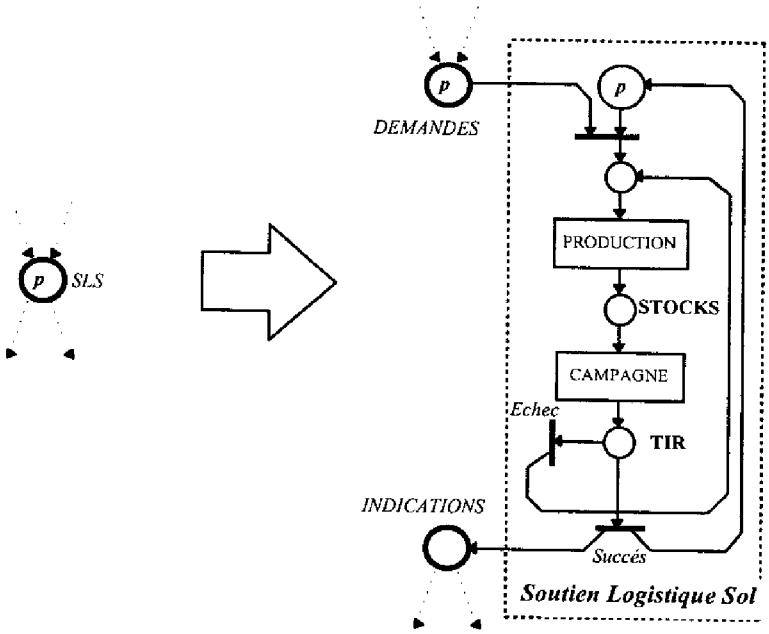


Figure V.6: Modèle Soutien Logistique Sol

Modèle Global

Le modèle global enfin obtenu est présenté par la Figure V.7. On retrouve la décomposition *Segment Spatial / Soutien Logistique Sol* où un seul modèle décrivant le processus d'initialisation et de maintenance est composé avec *p* modèles représentant le comportement des satellites sur chacun des plans d'orbite. Un seul de ces réseaux est décrit ici. Comme chaque plan d'orbite présente le même comportement, l'utilisation des réseaux colorés permettrait de se contenter d'un seul modèle *PLANi* avec ses interfaces. L'appartenance d'un satellite à tel ou tel plan étant caractérisée par la « couleur » du jeton donnant son comportement.

Deux types de transitions ont été différenciés: celles représentées par un rectangle plein sont des transitions de synchronisation, elles correspondent aux transitions immédiates du modèle RdPTS utilisé pour la phase d'évaluation. Les transitions illustrées par un rectangle vide représentent, elles, un processus de durée non nulle. Ce sont les transitions temporisées du modèle RdPTS.

Notons que, moyennant la connaissance minimale nécessaire à la compréhension d'un réseau de Petri, cette représentation est bien plus claire que la description textuelle du scénario. L'effort de modélisation permet en effet de lever certaines ambiguïtés de spécifications.

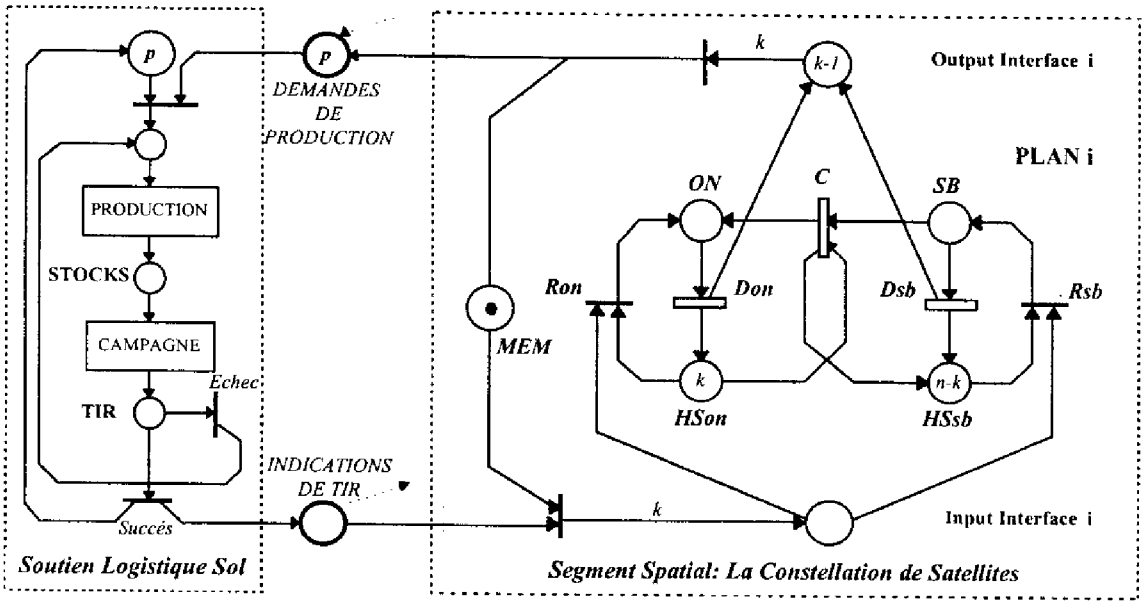


Figure V.7: Modèle Global du Scénario Simplifié

V.4. Scénario Détaillé

Sur la base du modèle précédemment construit, nous allons enrichir progressivement les spécifications, et donc les modèles, décrivant le déploiement et la maintenance des satellites de la constellation. Ceci est réalisé en trois temps.

Tout d'abord la communication entre le modèle *Soutien Logistique Sol* et le modèle *Segment Spatial* est adaptée de façon à permettre le stockage de satellites au sol pour répondre plus promptement aux besoins de chaque plan. Le modèle *Soutien Logistique Sol* est par la suite enrichi pour prendre en compte le partage des ressources *chaînes de production* et *pas de tir* identifiées lors de la description du système. Enfin, le modèle de chaque plan est développé pour décrire la fin de vie des satellites nominaux et le moyen de s'en prévenir en anticipant les demandes en satellites correspondantes. Cette possibilité nécessite également de revoir le modèle d'interface de sortie.

Les modifications et enrichissements des modèles déjà présentés seront mis en évidence graphiquement: la couleur noire sera appliquée à toute partie de réseau modifiée ou augmentée tandis que le gris correspondra à des parties décrites auparavant.

V.4.1. Stockage au Sol des Satellites

Dans le scénario simplifié, seulement p demandes initiales de satellites sont effectuées et aucun stockage au sol des satellites n'est prévu. Ainsi, une fois le déploiement effectué, lorsque se produit une défaillance sur l'un des plans, une demande de k satellites est émise vers le *Soutien Logistique Sol*. Cette demande ne peut être traitée que lorsque ces k satellites ont été produits, intégrés, lancés et positionnés sur le plan demandeur. Il est possible de réduire la période d'indisponibilité sur un plan demandeur et stockant au préalable des satellites au sol. De nombreuses options de stockage sont possibles et celle que nous retenons ici suppose que s lots de k satellites sont, au plus, disponibles.

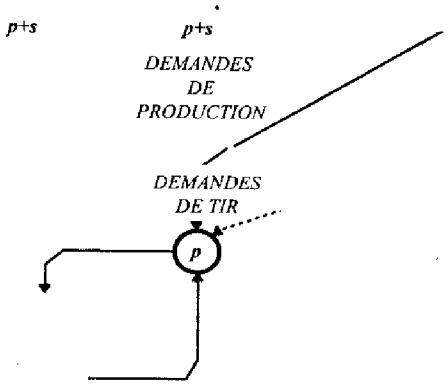


Figure V.8: Modification de la Communication entre le Modèle *Segment Spatial* et le Modèle *Soutien Logistique Sol*

Initialement *(p+s)* (au lieu de *s*) lots de satellites sont produits. Les *p* premiers sont utilisés pour remplir chaque plan et les *s* suivants sont stockés au sol.

Lorsqu'un plan a besoin d'un lot de satellites une demande de lancement est effectuée en parallèle d'une demande de production. Si un lot est disponible il est utilisé sinon il faut attendre la prochaine production.

Le réseau de la Figure V.8 prend en compte ces modifications. Le marquage initial de la place *DEMANDES DE PRODUCTION* passe à *(p+s)*, une place *DEMANDES DE TIR* a été rajoutée afin de ne pas déployer les satellites prévus pour le stockage au sol. C'est pourquoi son marquage initial vaut *p*. Cette place contrôle donc le tir de la transition *CAMPAGNE*. Elle est marquée à nouveau lorsqu'un tir échoue afin de le recommencer dans les plus brefs délais.

V.4.2. Partage de Ressources Sol

Jusqu'à présent, la production aussi bien que le stockage des satellites et les campagnes lanceur étaient modélisées par de simples transitions. Nous allons maintenant affiner ces processus.

L'obtention de lots de *k* satellites prêts à être lancés nécessite en fait deux actions en parallèle: d'une part la production des *k* satellites qui met en oeuvre le partage des ressources *c* chaînes de production et, d'autre part, la commande de lanceur. On suppose que chaque lanceur est à même de déployer *k* satellites.

Le stockage (*k* satellites, *1* lanceur) et la campagne de tir sont un peu plus complexe. En effet, lorsqu'un couple (*k* satellites, *1* lanceur) est disponible, deux options se présentent. Soit il n'y a plus de demande de tir de la part du segment spatial et le couple est stocké, soit ce n'est pas le cas et le couple est immédiatement lancé.

Le réseau de la Figure V. 9 détaille ces comportements.
Le tir de la transition *Début Production / Commande* lance en parallèle la production des *k* satellites et la réservation d'un lanceur. Si cette dernière est de durée supérieure à celle de la production des satellites, c'est la tâche la plus longue uniquement lorsque les ressources *c chaînes de production* sont disponibles. Autrement l'attente de la libération des ressources peut rendre la production des satellites de durée plus longue. Une fois que *k* satellites sont produits et qu'un lanceur est disponible, la transition *Fin Production / Commande* peut être franchie.
Si un tir est demandé par le segment spatial (marquage de la place *DEMANDES DE TIRS*) le nouveau jeton est utilisé pour immédiatement débuter une campagne lanceur en utilisant les *1* ressources *pas de tirs*. Dans le cas contraire (test à zéro de la place *DEMANDES DE TIRS*), le jeton est mis de côté dans la place *Stocks Satellites*. Si une nouvelle demande survient, il pourra immédiatement être utilisé (moyennant la disponibilité d'un pas de tir).
Le marquage initial des places *Chaînes de production* et *Pas de tir* indique que ces ressources sont disponibles.

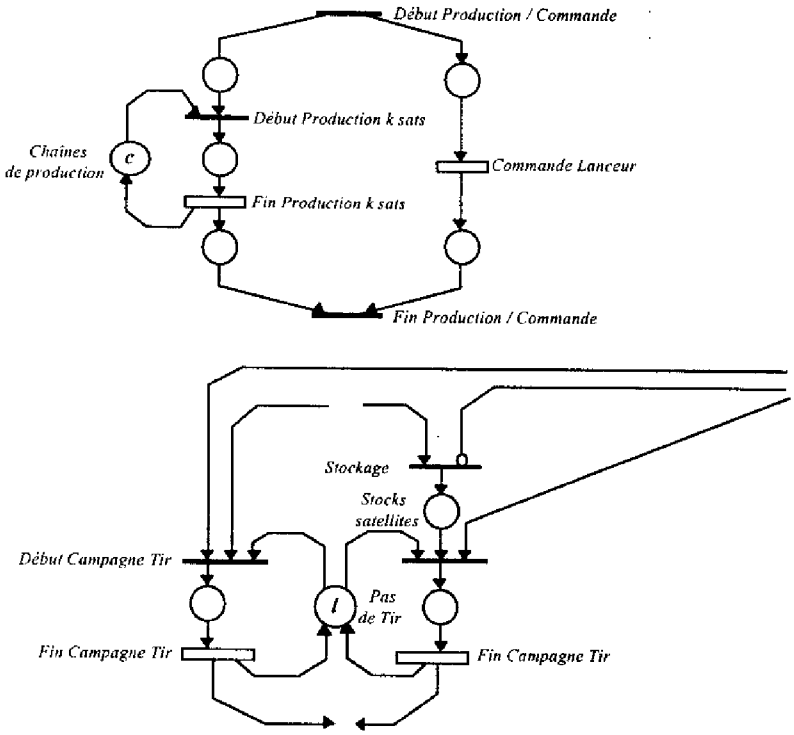


Figure V. 9: Partage de Ressources pour le Modèle *Soutien Logistique Sol*

V.4.3. Fin de Vie des Satellites Nominaux

Dans le scénario simplifié, on a supposé que les satellites devenaient hors d'état de marche uniquement suite à une défaillance. On considère maintenant que les satellites nominaux ont une durée de vie limitée, conditionnée par la consommation en ergol (cf. V.2).

Afin d'anticiper la demande de satellite résultante, on décompose en *4* étapes la vie des satellites actifs qui ne subissent pas de défaillances:

- La première correspond à la période normale de service,
- La seconde débute lorsque la durée de vie restante du satellite est égale à la durée normale de production et de tir d'un nouveau lot,
- La troisième débute lorsque cette durée de vie n'est plus égale qu'à la durée de campagne lanceur,
- Et enfin la dernière correspond à la mort d'un satellite.

Cette décomposition est modélisée dans le réseau de la Figure V.10. Les trois premières étapes sont représentées par les places *ON1*, *ON2* et *ON3*. A chacune de ces étapes, le satellite peut soit parvenir à l'étape suivante (tir d'une transition *step*), soit subir une défaillance (tir d'une transition *Don*). L'état hors service est toujours décrite par la place *HSon* .

Le marquage initial du réseau est inchangé et indique toujours qu'aucun satellite nominal ou de secours n'a été déployé.

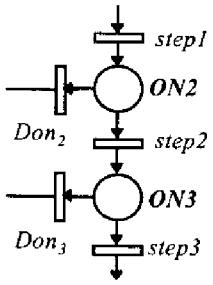


Figure V.10: Fin de Vie des Satellites Nominaux pour le Modèle *Plan d'Orbite*

Les événements liés aux changements d'étape ainsi qu'aux défaillances conditionnent les demandes de production et de tir de satellites requises par un plan. Ces demandes, comme on l'a vu, sont gérées par un modèle appelé *Interface de Sortie*. Afin de représenter l'anticipation de fin de vie, ce modèle a besoin d'être enrichi.

Jusqu'à présent, une demande de production et de tir est effectuée lorsque le plan a un satellite nominal défaillant et ne dispose plus de satellites de secours. De plus, ces demandes sont faites simultanément. Maintenant, on dissocie ces deux types de commandes afin de lancer les demandes de production avec anticipation sur les demandes de tir.

La Figure V.11 montre les liaisons entre les modèles d'interface et le modèle décrivant le comportement des satellites sur un plan d'orbite. De ce dernier modèle seuls les sommets intervenant dans les liaisons ont été rappelés. Le modèle d'*Interface d'Entrée* reste inchangé. Le modèle d'interface de sortie dispose maintenant de deux places P (pour production) et L (pour lancement). Chacune d'entre elle collecte les besoins du plan en satellites. Lorsque ces besoins sont égaux à un lot de k satellites, la demande correspondante est effectuée.

Le principe de réalisation des demandes est le suivant: tout satellite au cours de sa vie ne peut effectuer qu'une seule demande de production et qu'une seule de lancement. Ces dernières sont simultanées lorsqu'un satellite défaille alors qu'il était en veille ou bien en première phase de vie. Ces demandes sont dissociées dès lors qu'un satellite a atteint sa deuxième phase de vie. En effet le passage à cette deuxième phase engendre automatiquement une demande de production. La demande de tir est alors effectuée lorsque le satellite défaille ou atteint sa troisième phase de vie. C'est ce que représente la communication entre le modèle d'un plan et celui de son interface de sortie.

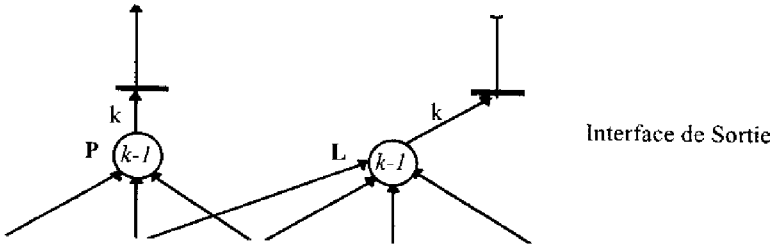


Figure V.11: Modèle *Interface de Sortie* Amélioré

V.5. Evaluation

Nous présentons dans cette section quelques résultats typiques de simulation des modèles construits dans la section précédente. Après avoir justifié le recours à la simulation pour ce type de réseau, on fixe les paramètres discrets, temporels et stochastiques du système afin d'obtenir des résultats sur les grandeurs étudiées.

V.5.1. Calcul Markovien ou Simulation ?

Lors d'une étude de Disponibilité, la spécification finale du système n'est pas obtenue directement mais elle est le résultat d'une démarche itérative impliquant plusieurs protagonistes. Afin de maîtriser la complexité du système, sa description est affinée progressivement. Ainsi, les premiers modèles peuvent être relativement simples pour permettre d'une part la vérification de certains comportements logiques et d'autre part la validation des résultats quantitatifs par comparaison entre le calcul Markovien et la simulation. Dans notre exemple, les modèles correspondant au scénario simplifié d'initialisation et de maintenance de la constellation ont pu être analysés qualitativement et quantitativement (non exposé ici) et fournir une bonne base pour l'affinement des hypothèses.

Rapidement cependant, il devient difficile de conserver une approche analytique pour l'évaluation des grandeurs observées. Ceci est particulièrement vrai lorsque le système est très fortement distribué et qu'il ne présente pas un comportement répétitif. Or c'est bien le cas d'une constellation de satellites dont le nombre d'états est rapidement considérable (plus de 160000 uniquement pour le modèle *Segment Spatial* du scénario simplifié pour les paramètres $\{p, n, k\}$ fixés à $\{3, 3, 2\}$) et qui présente plusieurs phases de fonctionnement: déploiement, entretien, remplacement suite à fin de vie. D'autre part, si le modèle Markovien peut s'appliquer correctement pour la défaillance des satellites, il n'est pas du tout envisageable pour décrire les processus de maintenance ou de durée de vie des satellites.

C'est pourquoi, même si les durées de traitements sont importantes et les résultats approximatifs, seule la simulation de tels réseaux permet de fournir des estimations des grandeurs comme la Disponibilité. L'évaluation se base donc sur la simulation de modèles RdPTS défini au cours du chapitre précédent.

V.5.2. Système Evalué

La robustesse de la modélisation par réseaux de Petri a permis de construire des modèles assez génériques qui sont valables quel que soient les valeurs des paramètres discrets *p, s, n, k, c* et *l*. D'autre part seule la distinction a été faite entre les processus immédiats ou synchronisation (rectangle plein) et ceux qui nécessitaient une certaine durée (rectangle vide). Une vaste étude de sensibilité peut donc être menée sur les paramètres discrets, temporels et stochastiques afin d'optimiser le système sans pour autant remettre en question les modèles. Tel n'est pas ici notre objectif, mais dans le cadre d'une étude réelle, cet aspect des choses est particulièrement intéressant.

Les tableaux suivants présentent les paramètres que nous avons retenus pour illustrer certaines sorties de l'évaluation des modèles.

- Paramètres discrets: la valeur de ces paramètres correspond aux jetons du même nom indiqué sur les réseaux.

Paramètre	*Valeur*	*Commentaire*
	SEGMENT SPATIAL	
p	*3*	*Nombre de plans d'orbite*
k	*2*	*Nombre de satellites actifs par plan*
n	*3*	*Nombre max de satellites par plan*
	SOUTIEN LOGISTIQUE SOL	
c	*1*	*Nombre de chaînes de production*
l	*1*	*Nombre de pas de tir*
s	*1*	*Nombre de stocks (k satellites, 1lanceur) max au sol*

- Paramètres Temporels Stochastiques: ils correspondent aux paramètres de loi associées aux transitions temporisées (rectangle vide) des réseaux présentés.

Transition	*Distribution*	*Paramètre*	*Commentaire*
	SEGMENT SPATIAL		
Don_i	$Exp(\lambda)$	$\lambda=1.15e\text{-}5$	*Fiabilité satellite actif de 0.6 à 5 ans*
Dsb_i	$Exp(\lambda')$	$\lambda'=1.15e\text{-}6$	$\lambda'=\lambda/10$
step1	$Dirac(\theta_1)$	$\theta_1 = 8 - T_{prod} - T_{camp}$	*1ère phase en années*
step2	$Dirac(\theta_2)$	$\theta_2 = T_{prod}$	*2ème phase en années*
step3	$Dirac(\theta_3)$	$\theta_3 = T_{camp}$	*3ème phase en années*
C	$Dirac(T_{change})$	$T_{change} = 2$ *mois*	*Activation d'un satellite de secours*
	SOUTIEN LOGISTQUE SOL		
Fin Prod	$Dirac(T_{prod})$	$T_{prod} = 12$ *mois*	*Durée de production 2 satellites*
Comm Lanceur	$Dirac(T_{comm})$	$T_{comm} = 18$ *mois*	*Durée de commande lanceur*
Fin Campagne	$Dirac(T_{camp})$	$T_{camp} = 2$ *mois*	*Durée campagne lanceur*

- Paramètre Stochastique: c'est la distribution de probabilité associée aux transitions en conflit *Echec* et *Réussite* qui caractérisent le tir d'un lanceur. On fixe *p(Echec)* à *0.1*, donc *p(Réussite)* à *0.9*.

Remarques

La même distribution a été associée aux transitions caractérisant la défaillance de satellites nominaux. Cela est possible uniquement car la distribution exponentielle est une loi sans mémoire. Si une autre transition avait été utilisée, il aurait fallu prendre en compte le temps passé dans les premières phases de vie.

Le conflit possible entre les transitions *Rec* (cf. Figure V.5) chargées de prélever pour un plan les *k* satellites nouvellement disponibles, n'a pas été probabilisé. Le comportement par défaut (sélection aléatoire uniforme de la transition) est satisfaisant dans la mesure où l'on ne souhaite pas favoriser un plan plutôt qu'un autre.

V.5.3. Paramètres de Simulation et Grandeurs Observées

La simulation des modèles a été réalisée avec le logiciel MISS-RdP présenté au cours du chapitre précédent. L'horizon de simulation choisi est de *20* ans ce qui est compatible avec la durée de service d'un tel système.

10000 histoires sont nécessaires pour avoir une précision de résultats suffisante.

La Disponibilité du *Segment Spatial* est observée à partir de l'état suivant:

$$E = \left\{ \sum_{j=1}^{p} \left[M(ON_1^j) + M(ON_2^j) + M(ON_3^j) \right] >= k \times p \right\}$$

En effet, l'étude des invariants de places du réseau de chaque plan d'orbite nous donne:

$$M(ON_1^j) + M(ON_2^j) + M(ON_3^j) + M(HS^j) = k$$

Donc, pour qu'aucun satellite actif ne soit défaillant (condition de Disponibilité d'un plan) il faut et il suffit que $M(ON_1^j) + M(ON_2^j) + M(ON_3^j) = k$. Appliqué à chacun des p plans, on retrouve la condition exprimée par l'état E qui caractérise la Disponibilité du *Segment Spatial*.

D'autres grandeurs peuvent être observées comme la Disponibilité dégradée (tolérance de la perte d'un certain nombre de satellites), le nombre de tirs de certaines transitions comme le nombre de tir de la transition *Fin Campagne* qui caractérise le nombre de lancements de satellites.

V.5.4. Résultats

La Figure V.12 présente les résultats de Disponibilité Instantanée obtenus grâce à l'observation de l'état E défini ci-avant. La courbe fine correspond à la stratégie non planifiée, dont les modèles ont été décrits, tandis que la courbe en gras présente les résultats d'une stratégie planifiée (pour laquelle les satellites sont régulièrement produits et stockés au sol) étalonnée sur la précédente.

La date *T0* correspond à la fin du déploiement de la constellation. On constate que la courbe débute assez bas (autour de *0.5*) et remonte très vite. Cette remontée est due au tir supplémentaire que nécessitent certaines histoires pour déployer totalement les satellites. La courbe se stabilise par la suite au dessus de *0.73*. La fin de vie des premiers satellites survient après *8* ans (*96* mois). La stratégie d'anticipation de fin de vie est efficace puisqu'on ne constate pas de chute importante de Disponibilité à cette période (tout au plus quelques légers créneaux).

La stratégie planifiée établie à partir de ces résultats suit le même déploiement que la stratégie précédente. La différence est que, au lieu de produire les satellites en fonction des besoins, ces derniers sont réalisés et stockés, si le *Segment Spatial* n'en a pas besoin, tous les 18 mois. C'est ce qui explique les dents de scie de la courbe. Une telle cadence de production planifiée est beaucoup plus compatible avec les contraintes industrielles.

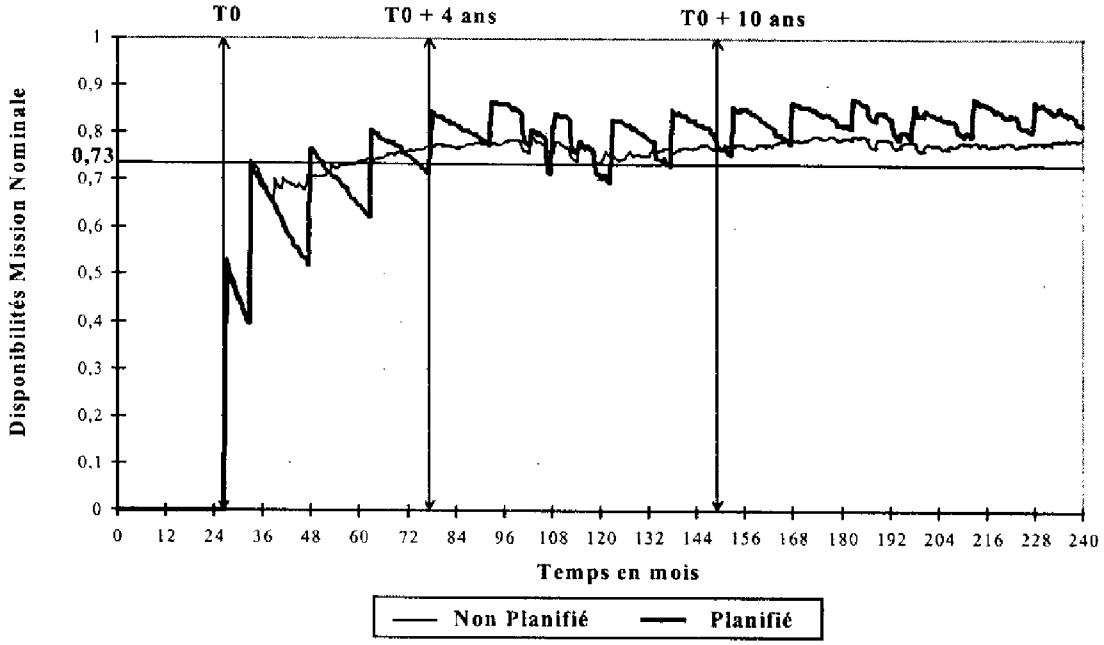


Figure V.12: Disponibilité Instantanée du Segment Spatial

Les courbes correspondantes aux intervalles de confiance à *95%* calculés pour *10000* histoires sont proposées sur la Figure V.13. Ils sont tout à fait satisfaisants au vu de la précision nécessaire pour ce type d'étude.

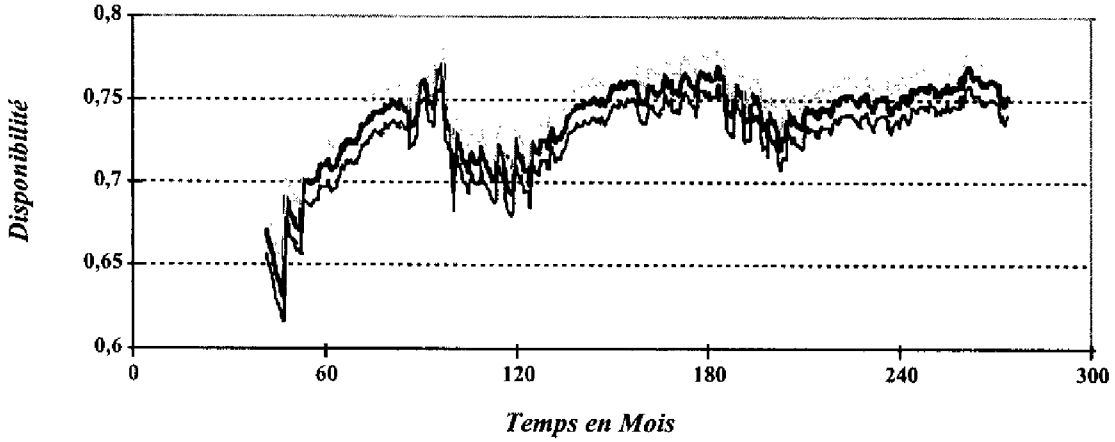


Figure V.13: Intervalles de Confiance obtenus pour *10000* histoires

Une observation plus fine de l'état de la constellation *4* ans après le déploiement est maintenant proposée. En effet, la distribution du nombre de tirs des transitions correspondant à la demande de production de satellites pour chacune des histoires simulées permet d'établir, à cette période, les besoins de la constellation en satellite. La Figure V.14 représente la distribution des histoires simulées en fonction du nombre de satellites commandés. Les histogrammes en blanc donnent la probabilité du nombre exact de satellites commandés (par exemple il y a environ *40%* de chances que *6* satellites soient commandés) tandis que les

histogrammes en gris donnent la probabilité cumulée (par exemple il y a presque *70%* de chances pour qu'au plus *6* satellites aient été commandés).

On constate que moins de *1%* des histoires se sont déroulées sans défaillance de satellites ou de lanceur (ce qui justifie, si besoin en était, ce type d'étude).

Ce type de résultats est particulièrement intéressant dans un contexte d'aide au dimensionnement. Il permet en effet de quantifier le risque que l'on prend avec telle ou telle configuration. Et cela donne des entrées précieuses notamment pour les études de coûts ou encore lors de négociations avec des partenaires.

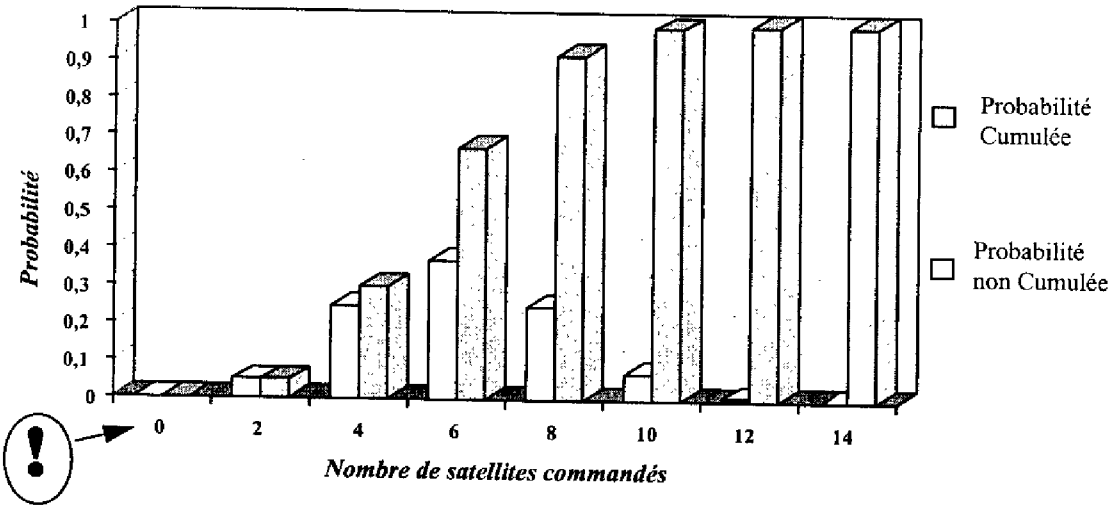


Figure V.14: Distribution du Nombre de Commandes en Satellites à (*T0+4*ans)

V.6. Conclusions

Le nombre de facteurs concourant au bon fonctionnement du segment spatial d'une constellation de satellites est tout à fait conséquent. L'optimisation de ces facteurs pour parvenir à un niveau spécifié de Disponibilité Opérationnelle est une tâche qui n'est pas triviale, qui débute tôt dans la conception du système et qui justifie un travail pluridisciplinaire. Il est tout à fait intéressant de disposer d'un même modèle intelligible par tous, pouvant intégrer ces différentes contraintes, et à partir duquel il est possible de mesurer l'impact des changements de telle ou telle hypothèse. Nous avons pu illustrer, au cours de ce chapitre, comment les réseaux de Petri répondent à cette attente là où d'autres méthodes de Sûreté de Fonctionnement étaient mise en échec aussi bien par manque de pouvoir descriptif (arbres de défaillance, blocs diagrammes de fiabilité) et de concision (graphes d'états) que par des possibilités limitées de traitement quantitatif.

La modélisation par réseaux de Petri a permis de décrire aussi bien les possibilités de défaillance des satellites que les processus d'initialisation, de maintenance et de remplacement. Les réseaux ont pu être construits progressivement et de façon suffisamment générique pour permettre une vaste étude de sensibilité afin d'optimiser le système.

La simulation, même si elle est gourmande en ressources informatiques, est l'unique solution envisageable pour un niveau de complexité des modèles comparable à celui présenté ici. Elle offre toute une palette de résultats qui dépassent largement la simple vérification de l'objectif de Disponibilité. De précieuses indications peuvent être fournies pour dimensionner les ressources de maintenance, ajuster les stratégies d'utilisation de ces ressources ou encore disposer d'arguments pour négocier avec des sous-traitants ou des sociétés d'assurances.

L'utilisation des réseaux de Petri pour ce type d'étude est ainsi légitimée. Toutefois, si grâce à une représentation graphique efficace la compréhension des modèles est assez aisée, la construction de ces derniers est, en revanche, une tâche non triviale qui nécessite du temps et une expérience significative dans ce domaine.

TROISIEME PARTIE

AIDE A LA MODELISATION

La deuxième partie de ce document a permis de justifier de l'intérêt des réseaux de Petri pour les études de Disponibilité Opérationnelle complexes. On a tout d'abord illustré les concepts clefs du modèle théorique des réseaux de Petri et des réseaux de Petri Stochastiques à partir de la déclinaison d'un exemple simple et bien familier des fiabilistes. Puis, on s'est appuyé sur une application plus concrète, propre au domaine spatial et pour laquelle les méthodes classiques étaient mises en échec.

L'intérêt des réseaux de Petri étant ainsi établi, l'objectif plus pragmatique de cette troisième partie est maintenant de contribuer à faciliter leur utilisation dans le contexte fortement contraint des avant-projets industriels. Disposant déjà d'un certain nombre d'outils efficaces pour le traitement quantitatif par calcul analytique ou simulation, nous avons centré nos efforts sur les problèmes de modélisation.

En effet, si à partir d'une représentation graphique efficace la compréhension des réseaux est assez aisée, la construction de ces derniers est, en revanche, une tâche non triviale qui nécessite du temps et une expérience significative. Des modèles construits à la hâte et sans méthode peuvent rapidement devenir des sacs de noeuds inextricables et donnant des résultats aisément contestables.

De plus, les études de Disponibilité Opérationnelle présentent la spécificité de mêler problèmes combinatoires (liens entre éléments) et description de processus discrets (initialisation, maintenance) ce qui alourdit considérablement les modèles que l'on peut construire.

Afin de simplifier et de systématiser la modélisation de systèmes complexes, nous suggérons une démarche en deux étapes.

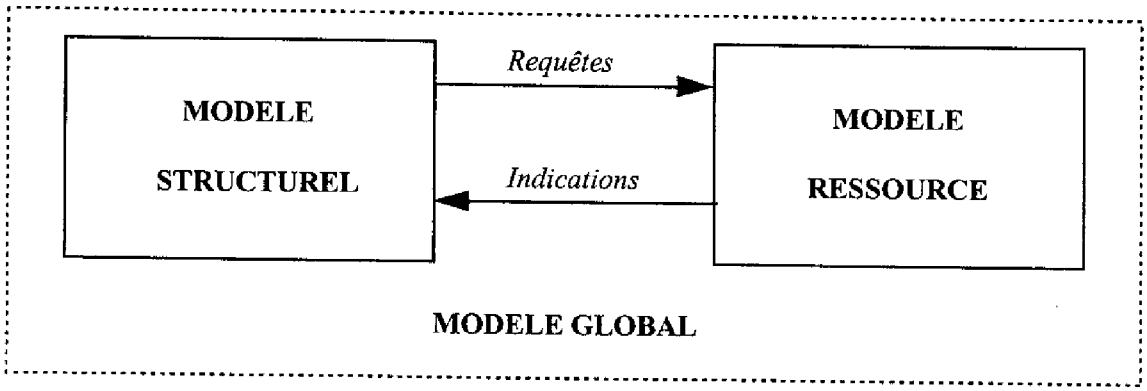
La première étape a pour but de décrire la structure du système c'est à dire les liens de dépendance du point de vue des défaillances, des fins de vie et des maintenances entre les éléments du système. C'est à ce niveau qu'on retrouve les problèmes combinatoires et qu'est décrit l'état des éléments. Ce modèle est nommé *Modèle Structurel*.

Si les défaillances et fins de vie peuvent être vues comme un phénomène indivisible, en revanche, les initialisations et maintenances sont le résultat de processus parfois complexes. Le rôle de la deuxième étape est de décrire ces processus et de les associer aux liens de maintenance préalablement établis. L'ensemble de ces processus sont regroupés dans un modèle appelé *Modèle Ressource*.

La communication entre ces deux modèles peut alors être vue comme une relation client-serveur où le client (*Modèle Structurel*) émet des requêtes au serveur (*Modèle Ressource*) afin d'obtenir des services d'initialisation, de maintenance ou de remplacement de ses éléments.

Le modèle global ainsi obtenu est alors un réseau de Petri que l'on peut traiter analytiquement ou par simulation.

Remarquons que c'est cette décomposition qui a été instinctivement suivie dans la modélisation de la constellation de satellites du chapitre précédent. En effet le modèle Segment Spatial représentait les liens de dépendance entre les éléments (*Modèle Structurel*) tandis que le modèle Soutien Logistique Sol décrivait tous les processus ressource nécessaires au déploiement et au maintien du système (*Modèle Ressource*).



Décomposition des Modèles

Si les réseaux de Petri sont particulièrement adaptés pour décrire les processus de la responsabilité du *Modèle Ressource*, en revanche, même s'il n'y a aucune impossibilité théorique, les liens de dépendance du *Modèle Structurel* sont beaucoup plus délicats et lourds à représenter.

L'objet de la troisième partie de ce mémoire est de présenter une approche pour la génération automatique de *Modèles Structurels* à partir d'un formalisme bien connu des fiabilistes: les Arbres de Défaillances. Le modèle obtenu, basé sur les réseaux de Petri, a été baptisé *Arbre de Défaillances Dynamique*.

Le chapitre VI définit l'objet théorique servant de support aux *Arbres de Défaillances Dynamiques*. Il s'agit d'une composition fortement contrainte de réseaux de Petri Synchronisés [Moalla 78] [David 92] : les arborescences de réseaux de Petri à synchronisation structurée.

Le chapitre VII est consacré aux *Arbres de Défaillances Dynamiques*. On présente les principes de génération automatique, les règles d'évolution ainsi que la mise en oeuvre informatique que nous avons réalisée.

Chapitre VI

Synchronisation Structurée d'une Arborescence de Réseaux de Petri

VI.1. Introduction

Représenter les liens de dépendance entre les éléments d'un système revient à décrire l'impact du changement d'état de chaque élément sur les autres éléments. Pour bâtir un *Modèle Structurel* (tel que nous l'avons présenté) il faut donc disposer d'un modèle capable de décrire de façon claire et structurée la propagation des effets d'une cause initiale.

Les règles d'évolution des différents modèles de réseaux de Petri que nous avons présentées jusque là sont toutes basées sur le franchissement des transitions les unes après les autres et de façon indépendante. Aucune règle n'est donnée sur le franchissement de séquences de transitions, si bien qu'une fois une transition franchie toute autre transition franchissable peut être candidate. Or la propagation d'effets, dans un réseau de Petri, ne peut être décrite que par une séquence de tirs de transitions. Et maîtriser cette propagation revient à contrôler le tir de séquences de transition.

La propagation d'effets peut être vue comme un mécanisme de synchronisation interne initié par un événement externe. Aussi la distinction entre les transitions liées aux événements externes (c'est-à-dire initiatrices de propagation) et celles mettant en œuvre ces propagations est incontournable.

Le modèle des réseaux de Petri Synchronisés [Moalla 78] [David 92] supporte cette distinction et établit des règles d'évolution basées sur les séquences de tirs de transitions. C'est pourquoi c'est le modèle que nous avons retenu pour la construction de *Modèles Structurels*. La section *2* de ce chapitre en présente sommairement les concepts clefs.

Afin de décrire les relations entre les éléments, il est nécessaire de composer les réseaux qui décrivent l'état de ces éléments. En effet, c'est cette composition qui met en oeuvre les mécanismes de propagation d'effets. La section *3* présente la composition par synchronisation interne de réseaux synchronisés.

Une structure arborescente est une bonne organisation d'éléments pour décrire des relations du type « *est composé de* ». Une structure arborescente de réseaux décrivant les états possibles d'éléments d'un système est donc un support efficace pour décrire la propagation d'effets. En décrivant les principes de synchronisation interne d'une telle arborescence, en imposant des contraintes aux réseaux de l'arborescence et en fixant ses règles d'évolution globales, on obtient un modèle particulièrement adapté à la construction de *Modèles Structure*. C'est l'objet de la dernière section de ce chapitre.

VI.2. Réseaux de Petri Synchronisés

Les réseaux de Petri Synchronisés sont des réseaux de Petri pour lesquels on différencie deux types de transitions : les transitions externes et les transitions internes. Les transitions externes sont étiquetées par un événement externe qui correspond à un changement d'état du monde extérieur. Une transition externe est franchie si elle est validée et dès que l'événement externe est occurrent. Les transitions internes sont elles étiquetées par l'événement toujours occurrent, elles sont donc franchies dès qu'elles sont validées. Sans prendre en compte le temps de façon explicite, cette distinction entre ces deux types de transition dans un réseau de Petri Synchronisé permet de décrire clairement les interactions d'un système avec le monde extérieur et les synchronisations internes qui lui sont propres.
Après avoir précisé leur définition, on détaille leur principe d'évolution ainsi que leurs propriétés caractéristiques.

VI.2.1. Définition

Définition VI.1 **Réseau de Petri Synchronisé**

Un *réseau de Petri Synchronisé* est un triplet $R_{Sync} = \langle R_E, E, Sync \rangle$ où :

- $R_E = \langle R, D, > \rangle$ est un réseau de Petri Etendu Simple c'est à dire tel que :
  $\forall t \in T,\ \forall p \in P,\ Pre(p,t) \leq 1$ *et* $Post(p,t) \leq 1$
- E est un ensemble d'événements externes
- $Sync : T \rightarrow E \cup \{e\}$ est la fonction de synchronisation (e est l'élément neutre du monoïde E^*)

❑

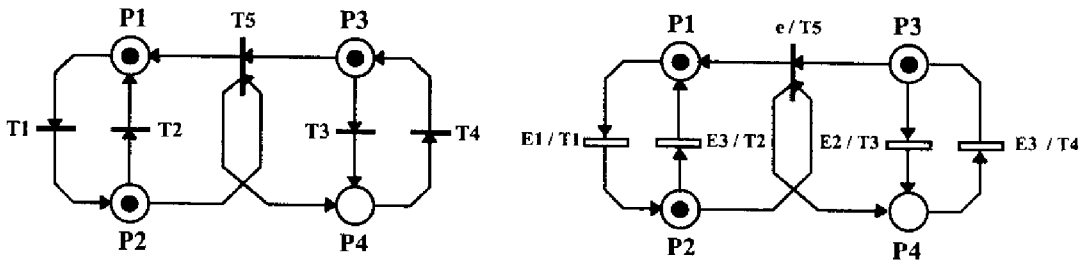
La fonction de synchronisation permet d'associer à chaque transition d'un R_{Sync} un événement. Cet événement peut être soit un événement externe, dans ce cas une des conditions de tir de la transition est liée à un facteur extérieur au modèle, soit l'événement e dit événement toujours occurrent qui permet de mettre en oeuvre des mécanismes de synchronisation interne.

On suppose que deux événements externes ne peuvent pas se produire simultanément.

On appelle *transition externe* une transition synchronisée sur un événement externe et *transition interne* une transition synchronisée sur *e*.
Une *transition externe* est représentée par un rectangle vide et une *transition interne* par un rectangle fin et plein.

Exemple

La Figure suivante présente un réseau de Petri (- a -) décrivant la redondance passive *2* parmi *3* avec réparation.



- a - Réseau de Petri Autonome ordinaire - b - Un réseau de Petri Synchronisé

Figure VI.1: Synchronisation d'un Réseau de Petri

Une synchronisation possible de ce réseau (- b -) est d'étiqueter ses transitions de la façon suivante :

- on associe aux transitions *T1* et *T3* les événements indépendants *E1* et *E2* correspondant à leur défaillance
- on associe aux transitions *T2* et *T4* le même événement *E3* correspondant à une réparation simultanée de tous les éléments défaillants. Notons que, contrairement au chapitres précédents, *E3* est maintenant l'événement « remplacement de tous les satellites défaillants » et non plus d'un seul satellite défaillant.
- et à la transition *T5* on associe l'événement toujours occurrent *e*. La commutation est alors vue comme une synchronisation interne se produisant dès qu'un élément primaire est défaillant et que l'on dispose d'au moins un élément en attente dans le mode secondaire.

❑

VI.2.2. Evolution

VI.2.2.1. Conditions d'Evolution

Les deux définitions qui suivent précisent les conditions d'évolution des réseaux de Petri Synchronisés, toujours basées sur les conditions de tir des transitions.

Définition VI.2 ***Réceptivité***

Une transition t d'un R_{Sync} est dite *réceptive* à un événement x pour un marquage M ssi :

- t est au moins 1-validée par M
- $Sync(t) = x$

❑

La réceptivité traduit l'aptitude d'une transition à être sensible à l'occurence de l'événement qui lui est associé. Elle devient franchissable lorsque cet événement survient.

Définition VI.3 ***Transition Franchissable***

Une transition t réceptive à un événement x d'un RSync est dite *franchissable* ssi x est occurrent.

❑

Une fois franchissable, la transition peut être tirée. Si c'est le cas, elle le sera immédiatement et selon les règles des réseaux de Petri Etendus.

Hypothèse d'évolution

Si une transition franchissable est q-validée, un seul franchissement de cette transition est autorisé pour une occurrence de l'événement externe associé.

VI.2.2.2. Séquence de Simulation Complète

Définition VI.4 ***Séquence de Simulation Complète***

S_k est une *séquence de simulation complète* (*SSC*) par rapport à un événement x, pour un marquage M, si elle remplit les conditions suivantes :

- S_k est une séquence de tirs à partir du marquage M, composée uniquement de transitions appartenant à $T_{x,M}$ ($T_{x,M} = \{t_1, t_2, ..., t_r\}$ est l'ensemble des transitions réceptives à x pour le marquage M).
- Toute transition de $T_{x,M}$ apparaît au plus une fois dans S_k.
- Toute séquence S_h obtenue en permutant les transitions de S_k est aussi une séquence de tirs à partir du marquage M.
- Il n'existe pas de séquence plus longue contenant toutes les transitions de S_k et remplissant les conditions 1, 2 et 3.

Propriété

Si S_k est une séquence de simulation complète, alors toute séquence S_h obtenue en permutant les transitions de S_k est aussi une séquence de simulation complète (S_k et S_h sont dites équivalentes).

❑

Exemple

On considère le réseau synchronisé de la Figure VI.1. Maintenant un événement externe *E4* est associé à la transition *T5*, la commutation n'est donc plus une synchronisation interne mais bien commandée par le monde extérieur. Le marquage a également été modifié.

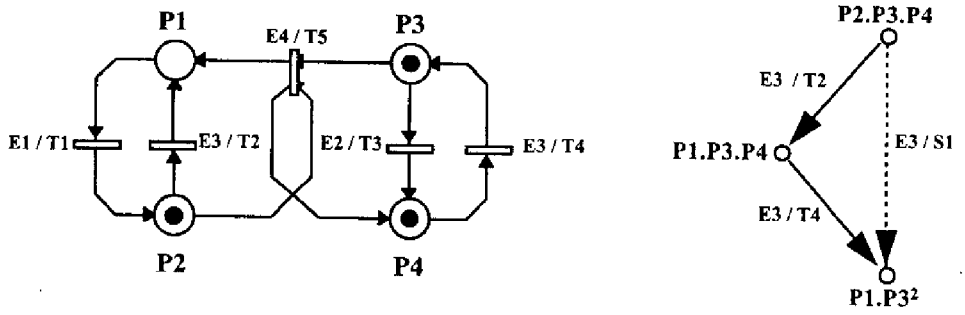


Figure VI.2: Exemple de Séquence de Simulation Complète

Les transitions *T2* et *T4* sont réceptives à l'événement externe *E3*. Sur occurrence de *E3*, on peut franchir *T2* puis *T4*. La séquence $S1 = E3 / T2.T4$ est une séquence de simulation complète, en effet :

- $T_{E3,M} = \{T2,T4\}$
- *T2* et *T4* apparaissent au plus une fois dans *S1*
- $S2 = E3/T4.T2$ est aussi une séquence de tirs possible à partir du marquage *P2.P3.P4*
- Il n'existe pas de séquence plus longue satisfaisant les conditions ci-dessus

❑

Remarque

Dans l'exemple précédent les deux séquences *S1* et *S2* mènent au même marquage. Ceci n'est pas une contrainte inhérente aux *SSC* mais correspond à la propriété de persistance que nous reprenons plus loin.

VI.2.2.3. Tir Itéré sur Occurrence d'un Evénement Externe

Définition VI.5 ***Tir Itéré sur occurrence d'un événement externe***

On appelle *Tir Itéré sur occurrence d'un événement externe x* pour un marquage *M*, le tir d'une *SSC* sur occurrence de *x*, suivi du tir d'une ou plusieurs *SSC* sur occurrence de *e*.

❑

Remarque

D'après l'hypothèse d'évolution, un tir itéré ne peut comporter qu'une seule *SSC* sur occurrence d'événement externe.

Notations

- Une *SSC* sur occurrence d'un événement externe (appartenant à *E*) est appelée *SSC* externe.
- Une *SSC* sur occurrence de e est appelée *SSC* interne.
- Si un tir itéré s'effectue en un nombre fini de *SSC*, on note $M \xrightarrow{x//\sigma} M'$ où σ est la concaténation des SSC successives utilisées, *M'* est appelé marquage stable final atteint.

- Une séquence de tirs itérés est notée $M \xrightarrow{\xi//} M'$ où ξ est la séquence d'événements externes engendrant la séquence de tirs itérés.

VI.2.2.4. Caractérisation des Marquages

Définition VI.6 ***Marquage Stable / Marquage Instable***

Un marquage M d'un réseau de Petri Synchronisé RS est dit *marquage stable* si, pour ce marquage, aucune transition de RS n'est réceptive à l'événement "toujours occurrent" e. Tout autre marquage est appelé *marquage instable.*

❑

Définition VI.7 ***Marquage Atteignable / Marquage Transitoire***

Un marquage M d'un réseau de Petri Synchronisé RS est dit *marquage atteignable* si c'est soit un marquage obtenu après le tir d'une *SSC*, soit le marquage initial. Tout autre marquage est appelé *marquage transitoire.*

❑

Remarque

Il ne faut pas confondre marquage atteignable et marquage accessible. Un marquage accessible est un marquage que l'on peut obtenir au cours de l'évolution d'un RSync, qu'il soit stable ou non, atteignable ou non.

Notations

- $A(RS; M_0)$: ensemble des marquages accessibles à partir du marquage M_0
- $A_S(RS; M_0)$: ensemble des marquages stables accessibles à partir de M_0
- $A_I(RS; M_0)$: ensemble des marquages instables accessibles à partir de M_0
- $A_A(RS; M_0)$: ensemble des marquages atteignables accessibles à partir de M_0
- $A_T(RS; M_0)$: ensemble des marquages transitoires accessibles à partir de M_0

On a alors pour un réseau synchronisé RS avec un marquage initial M_0 :

$$A = A_S \cup A_I = A_A \cup A_T$$

Remarques

Soit RS un réseau synchronisé et M_0 un marquage initial ,

- si $M \in A_S \cap A_A$, alors M est soit le marquage initial, soit le marquage résultant d'un tir itéré
- si $M \in A_I \cap A_A$, alors M est un marquage débutant une *SSC* interne

Pour un tir itéré sur occurrence d'un événement externe x, le marquage débutant la *SSC* externe est forcément un marquage stable. Les marquages transitoires de la *SSC* peuvent être des marquages instables mais pour lesquels on ne considère que les transitions réceptives à l'événement de la *SSC* interne. Si le marquage final de la *SSC* est un marquage atteignable instable alors on applique une ou plusieurs *SSC* internes jusqu'à mener, si possible, à un marquage atteignable stable.

VI.2.2.5. Algorithme d'Evolution

Algorithme **Interprétation d'un réseau de Petri Synchronisé**

Pas 1. Initialisation : 1) du marquage, 2) de l'échéancier des événements externes. Soit $x = e$. Aller au *pas 3.*

Pas 2. Considérer le premier instant θ de l'échéancier. Soit x l'événement se produisant à l'instant θ.

Pas 3. Déterminer l'ensemble des transitions franchissables sur occurrence de x. Si cet ensemble est vide, supprimer θ de l'échéancier et aller au *pas 2.*

Pas 4. Effectuer une *SSC*. Soit $x = e$. Aller au *pas 3.*

❑

Cet algorithme fait implicitement l'hypothèse que le nombre d'itérations des *pas 3 et 4* est fini, c'est à dire qu'à partir de tout marquage stable atteignable, toute occurrence d'événement externe conduit à un autre marquage stable atteignable en un nombre fini de *SSC*. On a une forte analogie avec le fait que dans le cas des réseaux de Petri Stochastiques Généralisés on ne doit pas avoir de séquence de tirs de transitions immédiates de longueur infinie.

VI.2.2.6. Exemple

L'exemple suivant illustre la notion de tir itéré sur occurrence de x et les divers types de marquages.

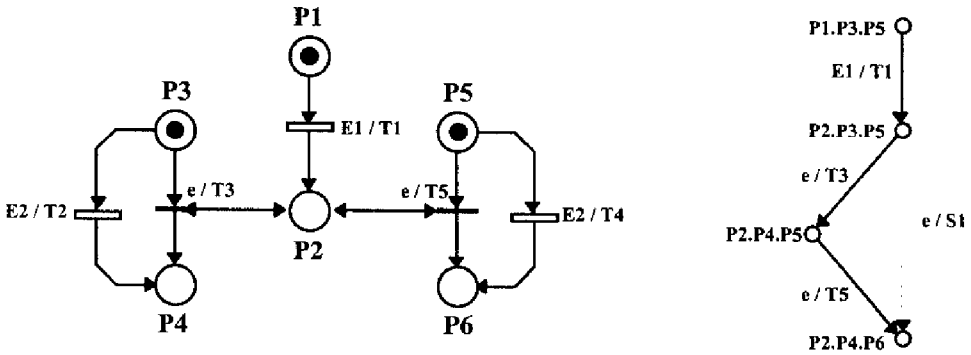


Figure VI.3: Exemple de Tir Itéré sur Occurrence d'un Evénement Externe

Remarque : Les arcs avec double flèche sont l'abréviation de deux arcs de même sommet et d'orientation opposée. Ils permettent de réaliser des tests de marquage.

Seule la transition *T1* est réceptive à l'événement *E1*, donc sur occurrence de *E1*, le tir de la *SSC* se résume au tir de *T1*. Ce dernier mène au marquage atteignable instable *P2.P3.P5*, on effectue une *SSC* interne (*S1 = e/T3.T5*) qui mène au marquage stable atteignable *P2.P4.P6*. Le marquage *P2.P4.P5* est un marquage transitoire instable.
Si les places *P3* et *P5* contenaient chacune *2* jetons, le tir itéré sur occurrence de *E1* serait le suivant :

1. Tir de la *SSC* externe : *S0 = E1/T1*
2. Tir de la *SSC* interne : *S1 = e/T3.T5*
3. Tir d'une nouvelle *SSC* interne : *S2 = e/T3.T5*

VI.2.3. Promptitude et Persistance

Globalement, mis à part le caractère borné, les propriétés ne sont pas conservées entre un réseau de Petri et un réseau de Petri Synchronisé. C'est notamment le cas de la vivacité. Nous présentons toutefois deux propriétés particulièrement intéressantes pour caractériser l'évolution d'un réseau de Petri Synchronisé: la promptitude et la persistance.

La promptitude est une propriété particulière aux réseaux de Petri Synchronisés, elle permet de caractériser les tirs itérés sur occurrence d'un événement externe qui s'effectuent en un nombre fini de pas. Un tir itéré prompt mène donc obligatoirement à un marquage atteignable stable en un nombre fini de *SSC* internes.

Définition VI.8 ***Promptitude***

Un RSync $R_S =< R, E, Sync >$ est dit *prompt* pour un marquage stable M_0 si :

$\forall M$ marquage stable atteignable, $\forall x \in E$, tout tir itéré sur occurrence de x appliqué à M s'accomplit au bout d'un nombre fini de *SSC* internes. Il est dit *k-prompt* si ce nombre est toujours inférieur ou égal à k.

❑

La persistance est une propriété déjà définie pour les réseaux de Petri Autonomes. Elle caractérise le fait que, pour un ensemble de transitions franchissables pour un marquage donné, l'ordre de tir de ces transitions est sans incidence sur le marquage final atteint. Cette propriété peut s'étendre aux réseaux de Petri Synchronisés.

Définition VI.9 ***Persistance***

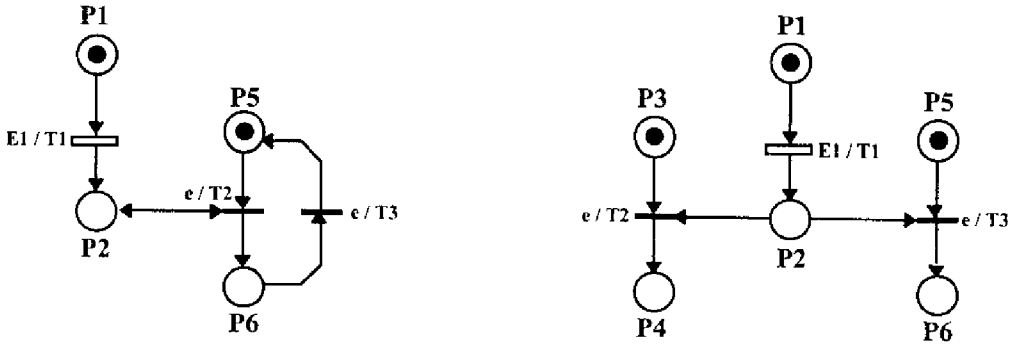
Le *RSync* est dit *persistant* pour M_0 ssi : $\forall M$ marquage stable atteignable, $\forall x \in E$, tout tir itéré sur occurrence de x appliqué à M aboutit au même marquage M'.

❑

Remarque

Un tir itéré composé uniquement d'une *SSC* externe peut ne pas être persistant.

Exemple



- a - Réseau non prompt

- b - Réseau non persistant

Figure VI.4: Promptitude et Persistance

Le réseau de la figure - a - est non prompt, en effet le tir itéré sur occurrence de *E1* peut être de longueur infini : séquence *T2.T3* toujours franchissable.
Le réseau de la figure - b - est non persistant car le tir itéré *E1/T1.T2* mène à un marquage différent de celui de *E1/T1.T3*.

❑

VI.3. Synchronisation Interne d'un Système de RSync

Les synchronisations internes au sein d'un *RSync* sont réalisées grâce aux transitions internes. On décrit ici la composition de plusieurs réseaux par le biais de tests de marquage associés aux transitions internes. C'est-à-dire que chaque transition interne d'un réseau est synchronisée sur l'état d'un ou plusieurs autres réseaux par le biais d'un test de marquage de ce ou ces réseaux. Ces tests de marquages très lourds graphiquement (double arcs) sont abrégés par un prédicat booléen étiquetant chaque transition interne. Ce prédicat est appelé événement interne. Après avoir décrit la macro et la micro-évolution d'un *RSync*, on précise la notion d'événements internes pour définir enfin la synchronisation interne d'un système de *RSync*.

VI.3.1. Macro et Micro-Evolution d'un *RSync*

VI.3.1.1. Réseau de Petri Synchronisé et Réseau de Petri Classique

L'évolution d'un réseau de Petri classique est purement asynchrone. C'est-à-dire qu'une seule transition est franchie à la fois et l'ensemble des transitions franchissables est réévalué après chaque tir.

Un réseau de Petri Synchronisé peut avoir un comportement similaire dans le cas particulier où il est totalement synchronisé (toutes ses transitions sont étiquetées par un événement externe) et où chacun des événements externes n'est lié qu'à une seule transition. En effet, dans ce cas, il n'est pas possible de franchir au même instant plusieurs transitions ce qui réduit toute *SSC* à une *SSC* sur occurrence d'un événement externe composée d'une seule transition. Les propriétés vérifiées par le réseau sous-jacent sont alors conservées pour le réseau synchronisé.

Cependant, dans le cas général, lorsque plusieurs transitions sont franchissables, un *RSync* évolue via le franchissement de *SSC*. Or, durant ce franchissement il n'y a pas réévaluation des transitions franchissables, elle ne se produit qu'une fois la *SSC* totalement franchie.

D'autre part, la classe des *RSync* autorise, s'ils sont possibles, les franchissements simultanés de plusieurs transitions d'une même *SSC*.

On s'éloigne donc du fonctionnement asynchrone des réseaux de Petri Classiques. Afin de préciser ce fonctionnement particulier, on définit la macro-évolution et la micro-évolution d'un *RSync*.

VI.3.1.2. Macro-Evolution et Micro-Evolution

Macro-évolution

L'ensemble des états stables atteignables d'un *RSync RS* pour un marquage initial M_0 $\left(A_{AS}(R_S; M_0) = A_A(R_S; M_0) \cap A_S(R_S; M_0)\right)$ correspond à l'ensemble des états initiaux ou terminaux des tirs itérés sur occurrence d'événement externe.
On peut, lorsque il est fini, le représenter sous la forme d'un graphe $GA_{AS}(RS; M_0)$ ayant pour ensemble de sommets l'ensemble des marquages de A_{AS}. Un arc orienté relie deux sommets M et M' s'il existe un tir itéré franchissable permettant de passer d'un marquage à l'autre. Les arcs sont étiquetés par l'événement externe et par la concaténation des *SSC* constituant le tir itéré. Ce graphe, en faisant abstraction des étapes des tirs itérés, caractérise la macro-évolution du *RSync*.

Micro-évolution

Si un *RSync* est prompt, un tir itéré est réalisé via le tir d'un nombre fini de *SSC*. Chaque *SSC* (excepté la dernière) mène dans un marquage atteignable instable.
A partir d'un marquage atteignable stable M, l'ensemble des marquages atteignables d'un tir itéré sur occurrence de x $(x \in E)$ peut être représenté sous la forme d'un graphe sans circuit $GA_A(RS; M; x)$ de source M et de puits M', le marquage stable atteint suite au tir itéré. Les autres sommets sont les marquages atteints instables. Un arc orienté relie deux sommets s'il existe une *SSC* du tir itéré les reliant.
Ce graphe caractérise alors la micro-évolution du *RSync* sur occurrence d'un événement externe pour un marquage stable atteint donné.

Exemple

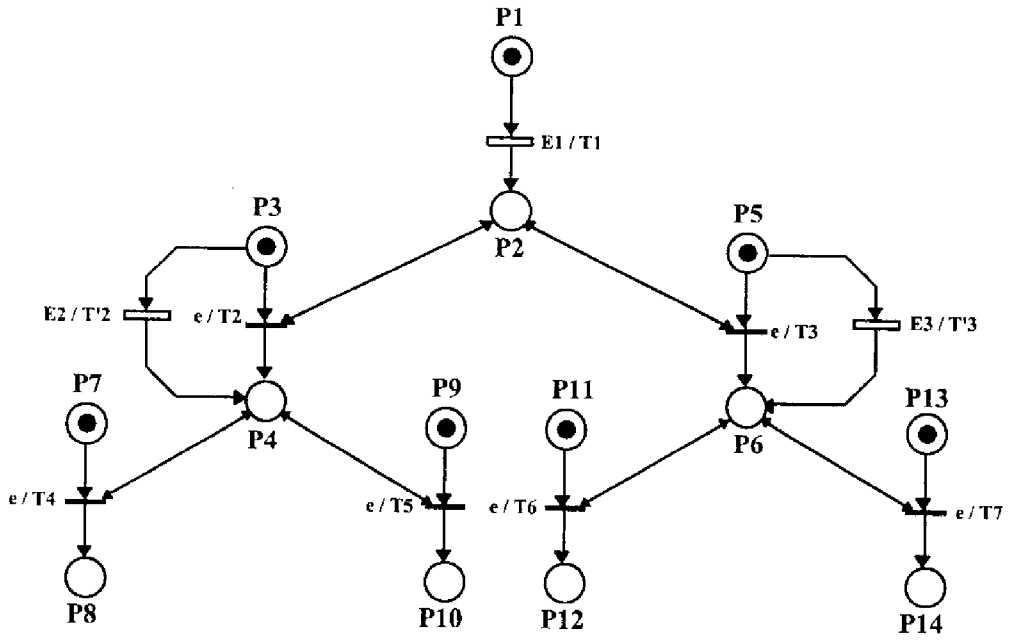


Figure VI.5: Exemple de Macro-Micro Evolution

Le *RSync RS* de la Figure VI.5 représente un système hiérarchisé constitué de *7* composants. Chaque composant est modélisé par un sous-réseau de *2* places (ex: *P1/T1/*P2 est l'élément sommet). La défaillance d'un élément d'un niveau entraîne celle des éléments de sa descendance.
Pour son marquage initial M_0, tous les éléments sont en bon fonctionnement
La macro-évolution de ce réseau est représentée par le graphe de ses marquages stables atteints (Figure VI.6). La concaténation des *SSC* d'un tir itéré est notée σ_i.

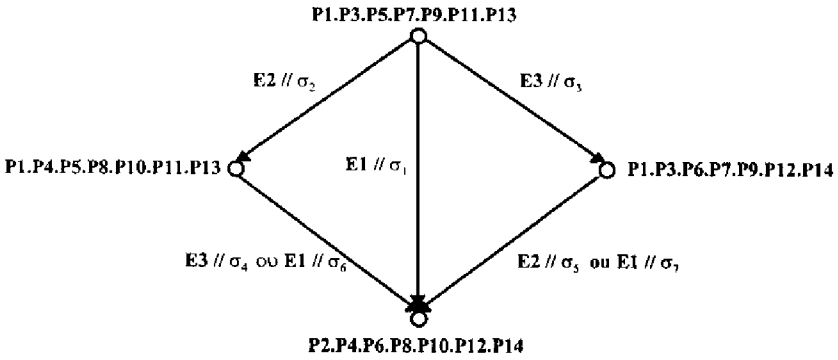


Figure VI.6: $GA_{AS}(RS; M_0)$

Le tir itéré sur occurrence de *E1* pour le marquage initial est constitué de *3 SSC* et passe par *2* marquages instables atteints. Cette micro-évolution traduit la propagation des défaillances de niveau en niveau.

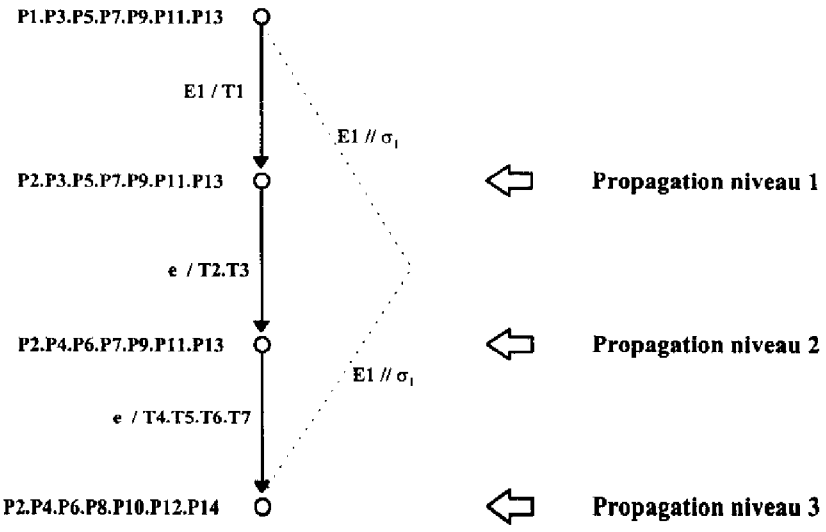


Figure VI. 7: $GA_A(RS; M_0; E1)$

□

Remarque

Un tir itéré peut être persistant sans pour autant que ses *SSC* le soient, c'est pourquoi le graphe sans circuits de micro-évolution peut comporter plusieurs chemins pour atteindre le marquage final du tir itéré.

VI.3.2. Evénements Internes

Comme on l'a vu, la micro-évolution d'un *RSync* débute toujours par une *SSC* externe. Cette dernière peut être suivie d'une ou plusieurs (voire d'une infinité de) *SSC* internes. Ces *SSC*, qui ne concernent que les transitions internes, sont totalement indépendantes de l'évolution du monde extérieur qui est "figé" durant la micro-évolution. C'est pourquoi elles constituent les mécanismes de *synchronisation interne* du *RSync*.

Au cours de la micro-évolution, le tir d'une transition interne n'est sujet qu'au marquage de ses places amonts. Notre but est de composer des *RSync* par synchronisation interne, c'est à dire enrichir la condition de tir d'une transition interne d'un réseau par un prédicat construit sur l'état d'un ou plusieurs autres réseaux. Ce prédicat est appelé *événement interne*, il permet de mettre en oeuvre des synchronisations internes entre plusieurs *RSync*. Précisons donc de façon plus formelle sa définition.

VI.3.2.1. Définitions

Notations

Soit $(RS;\ M_0)$ un *RSync* synchronisé marqué, si $Card(P) = n$ alors tout marquage peut s'écrire sous la forme : $M = P_1^{\mu_1}.P_2^{\mu_2}...P_n^{\mu_n}$ avec $\mu_i \geq 0$, nombre de jetons contenu par la place P_i. On note

- $|M|$: longueur de M c'est à dire le nombre de places marquées pour M
- $|M|_{P_i}$: nombre de jetons dans P_i pour le marquage M ($|M|_{P_i} = \mu_i$)
- $im(M) = \{p \in P \ / \ |M|_p > 0\}$: l'ensemble des places marquées

Associons à chaque place $p \in P$, la variable booléenne "place p marquée" $B(p)$:

$$B : P \longrightarrow \{0,\ 1\}$$

$$p\ \alpha\ B(p) \overset{Notation}{=} p = \begin{cases} 1 \text{ si } p \text{ est marquee} \\ 0 \text{ sinon} \end{cases}$$

On note $B(P)$ l'ensemble des variables booléennes associé aux places de l'ensemble P. Il constitue un ensemble de littéraux à partir desquels on peut construire des expressions logiques booléennes grâce au système algébrique constitué de $\{\{0,\ 1\},\ \neg,\ +,\ \times\}$.

Définition VI.10 ***Evénement Interne***

On appelle *Evénement Interne* sur un ensemble de places P toute expression booléenne bâtie sur $B(P)$ à partir de $\{\{0, 1\}, \neg, +, \times\}$. On note $EI(P)$ l'ensemble des événements internes possibles pour l'ensemble P.

❑

Remarque

Comme $B(p)$ est abrégé par p, $B(P)$ sera abrégé par P.

Lors de l'évolution d'un *RSync RS*, un événement interne peut se produire s'il existe au moins un état atteignable tel que le marquage des places de *RS* rend l'événement vrai. Seuls les états atteignables peuvent générer un événement interne car la liste des transitions constituant une *SSC* n'est évaluée qu'à partir d'un état atteignable.
Pour caractériser ces événements, on définit la fonction macro-état qui, à chaque événement interne, associe l'ensemble des marquages atteignables pour lesquels cet événement est réalisé.

Définition VI. 11 ***Fonction Macro-État***

$$\Psi : EI(P) \longrightarrow A_A(R_S ; M_0)$$

- $\Psi(P_i) = \{ M \in A_A(R_S ; M_0) \ / \ P_i \in im(M) \}$
- $\Psi(P_i \times P_j) = \Psi(P_i) \cap \Psi(P_j)$
- $\Psi(P_i + P_j) = \Psi(P_i) \cup \Psi(P_j)$
- $\Psi(\neg P_i) = A_A(R_S ; M_0) - \Psi(P_i)$

❑

Définition VI. 12 ***Evénement Interne Observable***

Soit $\varepsilon \in EI(P)$, ε est un *Evénement Interne Observable* ssi $\Psi(\varepsilon) \neq \varnothing$.
On note $EIO(RS, P, M_0)$, l'ensemble des événements internes observables sur P pour le réseau synchronisé *RS* avec le marquage initial M_0.

❑

VI.3.2.2. Evénements Internes et Tests de Marquage

Afin de composer des *RSync* par synchronisations internes, nous étiquetterons les transitions internes d'un réseau par un événement interne composé sur les événements internes d'autres réseaux. On peut penser qu'on étend alors le pouvoir de description des *RSync*. Il n'en est rien car à tout événement interne on peut faire correspondre une composition de tests de marquage exprimés en réseau de Petri par des boucles élémentaires. La Figure VI.8 montre cette équivalence.

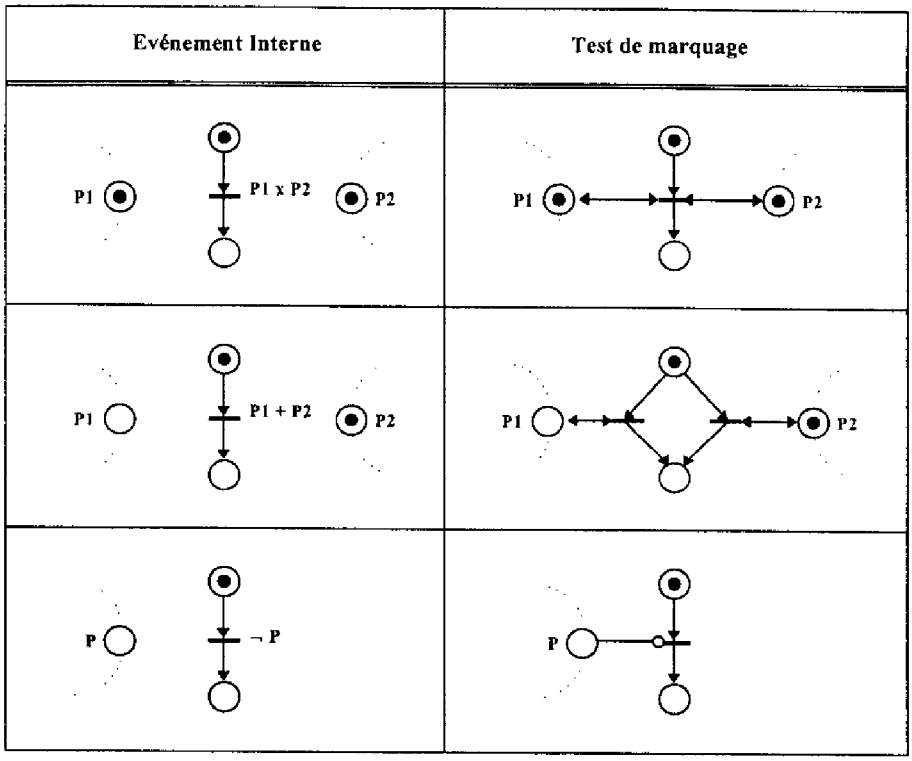


Figure VI.8: Equivalence entre Evénements Internes et Tests de Marquage

Remarques

- Le marquage de ces réseaux autorise le tir des transitions
- Le prédicat ¬ P correspond à un test de marquage par un arc inhibiteur.

VI.3.2.3. Evénement Interne et Evénement Externe

Un événement externe correspond à un changement d'état du monde extérieur, il peut donc être vu comme une impulsion. Ainsi, seules les transitions franchissables pour cet événement pourront être tirées et ce au plus une fois. C'est à dire que sur occurrence de cet événement une seule *SSC* liée à ce dernier peut être franchie. Une nouvelle occurrence de cet événement est nécessaire pour effectuer une nouvelle *SSC*.

Un événement interne, comme défini ci-avant, correspond à un marquage (ou une combinaison de marquages) d'un ensemble de places d'un ou plusieurs réseaux. Cet événement reste vrai tant que ce marquage (ou cette combinaison de marquages) reste vrai(e). Il n'est donc pas de nature impulsionnelle mais peut être représenté par un niveau. Le tir d'une *SSC* sur occurrence d'un événement interne ne "consomme" pas ce dernier, il est donc possible de franchir une nouvelle *SSC* sur cet événement.

VI.3.3. Définition

Définition VI.13 ***Synchronisation Interne d'un Système de RSync***

Soit Σ_{RS} un système de n *RSync* RSi $(i = 1, .., n)$. Pour chaque RSi on pose : $P_i = P_{Si} \cup \{P_i - P_{Si}\}$, $T_i = T_{Si} \cup \{T_i - T_{Si}\}$

On appelle $\{T_{Si}, P_{Si}\}$ l'interface de R_{Si} avec :
P_{Si} : ensemble de places de synchronisation permettant l'émission d'événements internes vers RS_j, $j = 1, .., n$ et $j \neq i$
T_{Si} : ensemble des transitions internes de R_{Si}.

La *Synchronisation Interne* de Σ_{RS} est réalisée par le biais des fonctions de synchronisation interne. Une fonction de synchronisation interne associe à chaque transition interne d'un réseau de Σ_{RS} un événement interne bâti sur les places de synchronisation des réseaux sur lesquels il est synchronisé.

$$Sync_{RSi} : T_{Si} \longrightarrow EI(\cup P_{Sj}) \quad \text{pour } i, j = 1, .., n \text{ et } i \neq j$$

❑

Remarque

Cette définition exclut, dans le réseau global issu de la composition, l'existence de transitions internes non synchronisées. C'est à dire que le franchissement de toute transition interne d'un réseau est conditionné par le marquage des places de synchronisation des autres réseaux.

Notations

- Chaque transition de T_{Si} synchronisée sur un événement interne sera étiquetée par l'expression booléenne représentant cet événement.
- On note $Sync(T_{Si})$ l'ensemble des événements internes sur lesquels sont synchronisées les transitions internes de RSi

La figure suivante illustre le principe de synchronisation interne de deux réseaux synchronisés.

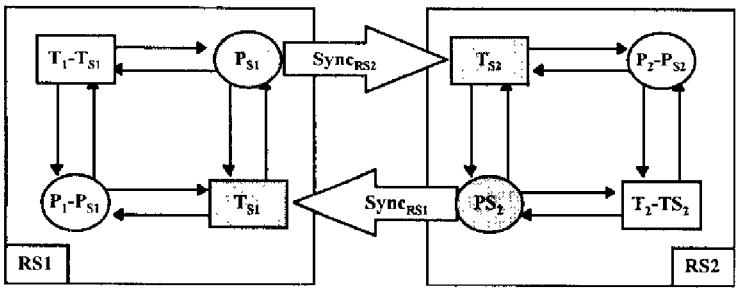


Figure VI.9: Composition de *2 RSync*

VI.3.4. Evolution

L'évolution d'un système de *RSync* synchronisé de façon interne est exactement celle du *RSync* global Σ_{RS} pour lequel les événements internes ont été remplacés par des tests de marquage entre réseaux selon les principes présentés auparavant. L'algorithme d'évolution est alors strictement celui énoncé pour les *RSync*.

L'utilisation des événements internes permet de conserver la décomposition d'un système en plusieurs réseaux, c'est à dire la modularité du modèle global. Les prédicats étiquetant les transitions internes d'un réseau indiquent alors les liens de dépendance de ce réseau avec le reste du système.

Si l'on observe l'évolution d'un seul réseau alors, le système auquel il appartient peut être vu comme un monde extérieur à lui sur lequel il est synchronisé via sa fonction de synchronisation interne.

Les notions de réceptivité et de franchissabilité peuvent alors être précisées pour les transitions internes.

Définition VI.14 ***Réceptivité d'une Transition Interne***

Soit *RSi* appartenant à un système de *RSync* synchronisé Σ_{RS}.
Soit $t \in T_{Si}$, t est *réceptive* à l'événement interne x pour le marquage M de Σ_{RS} ssi :

- t est au moins 1-validée par M
- $Sync(t) = x$

❑

Définition VI.15 ***Transition Interne Franchissable***

Soit *RSi* appartenant à un système de *RSync* synchronisé Σ_{RS}.
Soit $t \in T_{Si}$ une transition interne de *RSi* réceptive à x pour le marquage M de Σ_{RS}, t est franchissable pour M ssi x est occurrent, c'est à dire $M \in \Psi(x)$.

❑

VI.3.5. Exemple

Supposons deux réseaux *RS1* et *RS2* représentant le comportement de deux éléments E_1 et E_2. Chaque élément peut défaillir et être réparé, cependant la défaillance de E_2 entraîne celle de E_1 et la réparation des deux éléments est décidée lorsque E_1 est défaillant.

Le modèle qui suit (Figure VI.10) n'est certainement pas le plus simple mais permet de conserver la décomposition structurelle grâce aux deux réseaux composés par synchronisation simple.

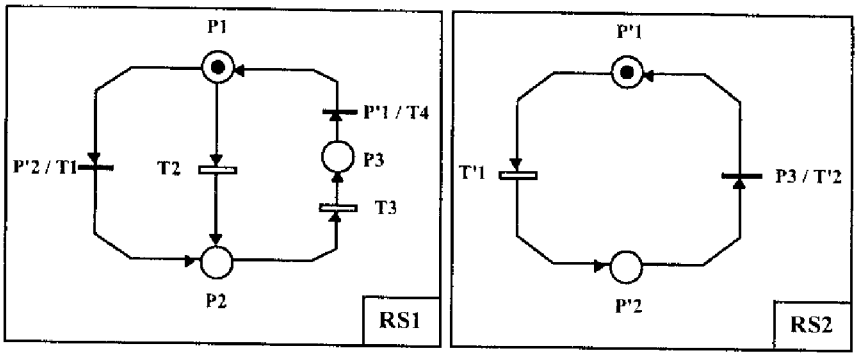


Figure VI.10: Exemple de synchronisation simple

Synchronisation de *RS1* sur *RS2* : $Sync_{RS1}(T1) = P'2$ et $Sync_{RS1}(T4) = P'1$.

Synchronisation de *RS2* sur *RS1* : $Sync_{RS2}(T'2) = P3$.

Exemple d'évolution : la transition *T1* est réceptive à l'événement interne *P'2*, supposons que la transition externe *T'1* soit tirée, alors *T1* devient franchissable car le nouveau marquage *M* est tel que *P'2* soit occurrent : $M = P1.P'2 \in \Psi(P'2)$.

VI.4. Synchronisation Interne d'une Arborescence de RSync (ARS)

Nous avons décrit précédemment le principe de synchronisation interne d'un système de *RSync* par le biais d'événements internes. On adapte ici cette synchronisation à un système de *RSync* organisé en une structure arborescente. On contraint l'échange d'événements internes à des réseaux de niveaux adjacents et de même parentée, cette forme de synchronisation interne est appelée synchronisation verticale. On introduit la notion de synchronisation horizontale qui autorise deux réseaux frères à partager une même transition externe. Enfin, en définissant des conditions sur la nature des réseaux de l'arborescence et de leurs synchronisations, on présente un algorithme de micro-évolution pour le système de *RSync* ainsi construit. Cette micro-évolution structurée est à la base du comportement des arbres de défaillances dynamiques qui font l'objet du chapitre suivant.

VI.4.1. Arborescence de *RSync* (*ARS*)

Une arborescence de *RSync* (*ARS*) est un système de *RSync* organisé en une structure arborescente. La relation liant les *RSync* est purement syntaxique et servira de support à la composition de ces réseaux par synchronisation interne.

VI.4.1.1. Définitions

Définition VI.16 ***Arborescence de RSync***

Une arborescence de *RSync* est définie par le triplet $ARS = <\Sigma_{RS}, RS0, \Gamma>$, avec :

- Σ_{RS} : ensemble fini de *RSync*
- $RS0$: racine de l'arborescence ($RS0 \in \Sigma_{RS}$)
- Γ : fonction successeur (ou descendance directe)

$$(\Gamma : \Sigma_{RS} \longrightarrow \Sigma_{RS} \times .. \times \Sigma_{RS})$$

❑

Remarque

- $\Gamma^0(\Sigma_{RS}) = id(\Sigma_{RS})$
- Pour $RS \in \Sigma_{RS}$ et $n \in N$, $\Gamma^n(RS)$ est l'ensemble des descendants de RS de $n^{ième}$ génération

Niveaux d'une ARS

Une *ARS* peut être décomposé en n niveaux ($n \in N$) : $\Sigma_{RS} = \Sigma_{RS0} \cup \Sigma_{RS1} \cup .. \cup \Sigma_{RSn-1}$ avec :

- $\Sigma_0 = RS0 = \Gamma^0(\Sigma_{RS0})$: unique réseau de niveau *0*
- $\Sigma_i = \Gamma^i(\Sigma_{RS0})$: ensemble de réseaux du niveau *i* ($i<n$)
- $n(RSk) = \Sigma_i$ ssi $RSk \in \Sigma_i$: niveau du réseau *RSk*

Descendance d'une ARS

Soit $RSk \in \Sigma_{RS}$, la descendance de RSk est définie par :
$desc(RSk) = \{RS \in \Sigma_{RS},\ \exists i > 0,\ RS \in \Gamma^{i}(RSk)\}$

Ascendance d'une ARS

De même, l'ascendance de $RSk \in \Sigma_{RS}$ est définie par :
$asc(RSk) = \{RS \in \Sigma_{RS},\ \exists i > 0,\ RS \in \Gamma^{-i}(RSk)\}$
(un seul ascendant par niveau inférieur)

Feuilles d'une ARS

Enfin les feuilles d'une *ARS* sont définies par :
$f(ARS) = \{RS \in \Sigma_{RS},\ \Gamma(RS) = \varnothing\}$

VI.4.1.2. Exemple

La figure suivante présente une *ARS* (dont les réseaux n'ont pas été détaillés) composée de *4* niveaux.

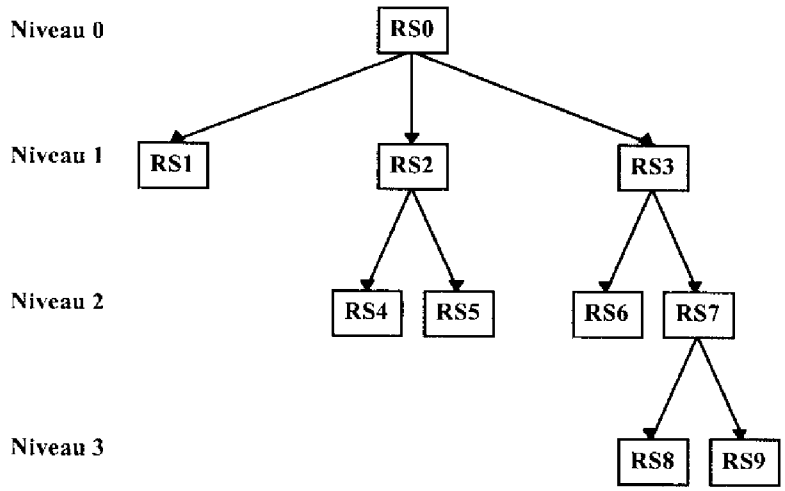


Figure VI.11: Exemple d'Arborescence de *RSync*

On a :

- $\Gamma(RS2) = \{RS4, RS5\}$, $\Gamma^{-1}(RS7) = \{RS3\}$
- $\Sigma_2 = \{RS4, RS5, RS6, RS7\}$
- $desc(RS3) = \{RS6, RS7, RS8, RS9\}$, $asc(RS8) = \{RS7, RS3, RS0\}$
- $f(ARS) = \{RS1, RS4, RS5, RS6, RS8, RS9\}$

VI.4.2. Synchronisation Interne d'une *ARS*

VI.4.2.1. Synchronisation Verticale

La synchronisation verticale est une synchronisation par le biais d'échange d'événements internes (ou tests de marquage) entre les réseaux d'une *ARS* qui s'appuie sur sa structure arborescente. C'est-à-dire que les événements internes associés aux transitions internes d'un réseau seront dépendants soit du marquage des réseaux fils, soit du marquage du réseau père.

Définition III-2 **Synchronisation verticale d'une arborescence de RSync**

Soit $ARS = <\Sigma_{RS}, RS0, \Gamma>$ une arborescence de RSync.

Pour chaque $RSi \in \Sigma_{RS}$, on pose : $P_i = P_{Si} \cup P_{Ei}$, $T_i = T_{Si} \cup T_{Ei}$ avec :

$P_{Si} = P_{Si}^{+} \cup P_{Si}^{-} \cup P_{Si}^{+-}$ *et* $T_{Si} = T_{Si}^{+} \cup T_{Si}^{-}$

$\{T_{Si}^{+}, T_{Si}^{-}, P_{Si}^{+}, P_{Si}^{-}, P_{Si}^{+-}\}$ est l'interface de *RSi* :

- T_{Si}^{+} : ensemble de transitions internes de *RSi* synchronisées sur des événements internes des fils de *RSi*,
- T_{Si}^{-} : ensemble de transitions internes de *RSi* synchronisées sur des événements internes du père de *RSi*,
- P_{Si}^{+} : ensemble de places de synchronisation permettant l'émission d'événements internes vers les fils de *RSi*,
- P_{Si}^{-} : ensemble de places de synchronisation permettant l'émission d'événements internes vers le père de *RSi*,
- P_{Si}^{+-} : ensemble de places de synchronisation permettant l'émission d'événements internes vers le père et les fils de *RSi*.

La *synchronisation verticale de l'arborescence ARS* est réalisée par le biais des fonctions de synchronisation internes de chaque réseau *RSi* de Σ_{RS}.

On pose $\Gamma(RSi) = \cup RSj$ et $\Gamma^{-1}(RSi) = RSk$,

$$Sync_{RSi}^{+} : T_{Si}^{+} \longrightarrow EI(\cup P_{Si}^{-} \cup P_{Si}^{+-}) \quad \text{et} \quad Sync_{RSi}^{-} : T_{Si}^{-} \longrightarrow EI(\cup P_{Sk}^{+} \cup P_{Si}^{+-})$$

❑

Remarques

- $T_{Si}^{+} \cap T_{Si}^{-} = \varnothing$
- $P_{Si}^{+} \cap P_{Si}^{-} \cap P_{Si}^{+-} = \varnothing$
- $T_{S0}^{-} = P_{S0}^{-} = P_{S0}^{+-} = \varnothing$ *et pour* $RSi \in f(ARS)$, $T_{Si}^{+} = P_{Si}^{+} = P_{Si}^{+-} = \varnothing$
- Les réseaux de même niveau ne sont pas synchronisés entre eux

Hypothèse importante

Deux transitions d'une *ARS* ne peuvent être synchronisées sur le même événement externe, donc tout tir itéré débutera par le tir d'une unique transition externe (*SSC* externe réduite à une transition).

Notations

Soit $RSi \in \Sigma_{RS}$:

- —⊖ : symbolise une transition de T_{Si}^{-}
- —⊕: symbolise une transition de T_{Si}^{+}

Exemple

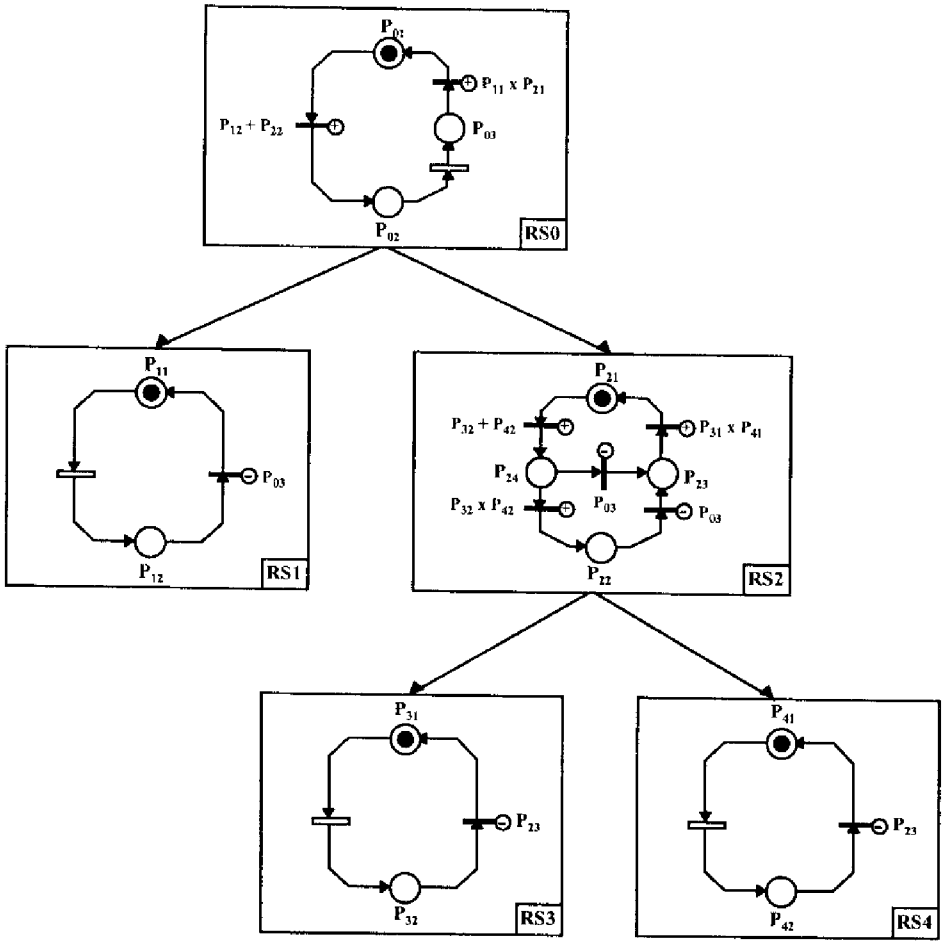


Figure VI. 12: Exemple de Synchronisation Verticale d'une Arborescence de *RSync*

La figure précédante décrit un système constitué de *3* composants (*C1, C3* et *C4*) modélisés par les réseaux feuilles *RS1*, *RS3* et *RS4*. L'association de *C3* et *C4* correspond au macro-

composant *E2* dont le comportement est décrit par le réseau *RS2*. Enfin, l'association de *E2* et *C1* représente le système entier décrit par le réseau racine *RS0*.
Si *C1* est défaillant, l'événement P_{12} est émis et le système devient défaillant (marquage de P_{02}).
D'autre part, si *C3* ou *C4* est défaillant, l'événement P_{32} ou P_{42} est émis et *E2* entre dans un mode dégradé (marquage de P_{24}).
Si *C3* et *C4* sont tous les deux défaillants, l'événement $P_{32} \times P_{42}$ est émis et *E2* devient défaillant. L'événement P_{22} est alors émis et le système devient défaillant.
Lorsque le système est défaillant, une réparation globale a lieu (tir d'une transition externe). L'événement P_{03} est émis vers *RS1* et *RS2*.
Si *C1* est défaillant, il est réparé.
Si *E2* est en mode dégradé ou défaillant, l'événement P_{23} est émis pour propager l'information de maintenance à ses composants. Ces derniers sont alors réparés et informent *E2* de cette réparation effective (événement $P_{31} \times P_{41}$) qui devient lui-même réparé. L'événement $P_{11} \times P_{21}$ est alors émis et le système devient lui même réparé.
On est ainsi revenu dans le marquage initial.

Remarque : Le principe de construction et de synchronisation d'une telle arborescence sera repris en détail dans la section suivante.

❑

VI.4.2.2. Synchronisation Horizontale

La composition des *RSync* au sein d'une *ARS* est réalisée jusqu'à présent par le biais d'événements internes échangés entre un réseau père et ses réseaux fils. On présente ici une nouvelle forme de composition structurelle pour laquelle deux réseaux frères partagent des transitions externes. C'est-à-dire que deux transitions externes partagées appartenant à deux réseaux frères deviennent une unique transition obtenue par fusion. Ce mode de synchronisation par rendez-vous, appelé synchronisation horizontale, permet de faire évoluer simultanément deux réseaux frères et s'avère particulièrement utile pour modéliser les commutations entre deux éléments dans le cas de la maintenance interne.

Définition VI.17 ***Synchronisation horizontale d'une ARS***

Soit $ARS = <\Sigma_{RS}, RS0, \Gamma>$ une arborescence de *RSync*.
Pour chaque $RSi \in \Sigma_{RS}$, on pose : $T_{Ei} = T_{Ei}^{P} \cup T_{Ei}^{NP}$ avec

- T_{Ei}^{P} : ensemble de transitions externes partagées de *RSi*
- T_{Ei}^{NP} : ensemble de transitions externes non partagées de *RSi*

La *synchronisation horizontale* de *l'ARS* est réalisée par le biais des fonctions de fusion $\mathcal{F}_{ij}$ qui pour deux réseaux frères *RSi* et *RSj* supprime deux transitions externes partagées t_i et t_j pour créer une transition t commune à *RSi* et *RSj* et obtenue par la fusion de t_i et t_j :
Pour *RSi* et *RSj* tels que $\Gamma^{-1}(RSi) = \Gamma^{-1}(RSj)$ on a :

$$\mathcal{F}_{ij} : T_{Ei}^{P} \times T_{Ei}^{P} \longrightarrow T_{E}$$

$\mathcal{F}_{ij}(t_i, t_j) = t$ telle que :

- $t^{\bullet} = t_i^{\bullet} \cup t_j^{\bullet}$ et ${}^{\bullet}t = {}^{\bullet}t_i \cup {}^{\bullet}t_j$

- $\forall p \in t^{\bullet}$, $Pré(p,t) = Pré(p,t_i)$ si $p \in P_i$
  $Pré(p,t_j)$ si $p \in P_j$
- $\forall p \in {}^{\bullet}t$, $Post(p,t) = Post(p,t_i)$ si $p \in P_i$
  $Post(p,t_j)$ si $p \in P_j$

❑

Remarques

- D'après cette définition, une transition partagée ne peut participer à plus d'une fusion (car la fusion supprime cette dernière) donc au plus deux transitions peuvent fusionner.
- La synchronisation horizontale ne concerne que des transitions externes.
- $\mathcal{F}_{ij}(t_i, t_j) = \mathcal{F}_{ji}(t_j, t_i) = t$

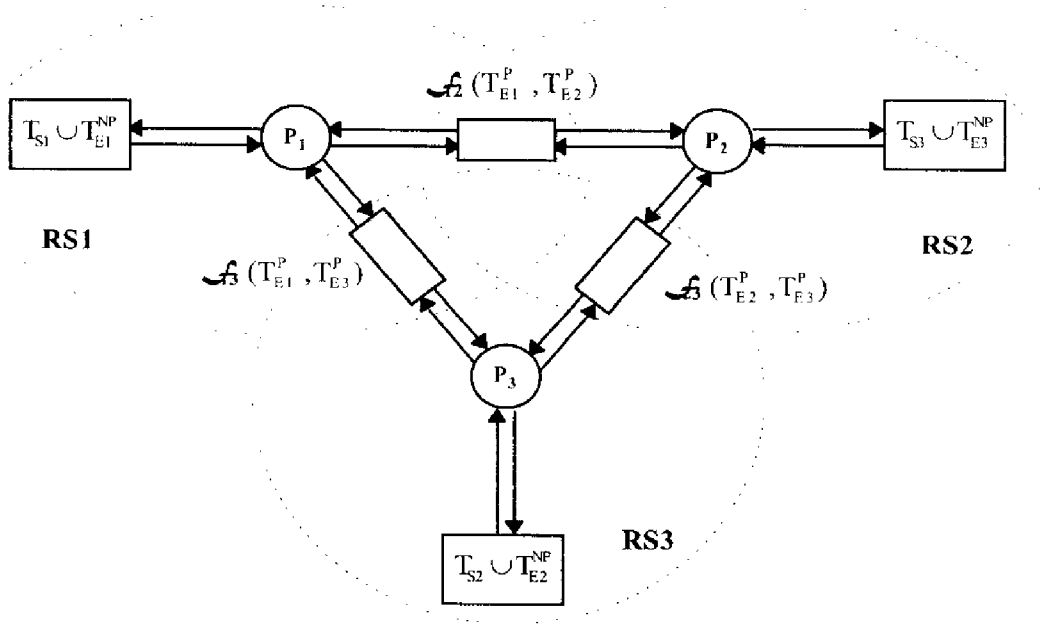


Figure VI.13: Synchronisation Horizontale de Trois Réseaux Frères

La figure précédente présente le synoptique de la synchronisation horizontale de trois réseaux frères.

Exemple

Soient deux réseaux *RS1* et *RS2*, chaque réseau représente les états possibles d'un composant. Un composant peut être en mode primaire (places *ON* et *HSon*) ou en mode secondaire (places *SB* et *HSsb*). Il peut défaillir ou être réparé (transitions *Don*, *Dsb*, *Ron*, *Rsb*) mais un composant en mode primaire qui devient défaillant peut également être remplacé par un composant en mode secondaire non défaillant. Dans ce cas une transition *Com.* est tirée. Ces transitions *Com.* sont issues de la fusion de transitions externes de *RS1* et *RS2*.

Par exemple, si le premier élément défaille (marquage de la place *HSon(1)*) alors la transition commune aux deux réseaux *Com. 12* est validée et le tir de cette dernière échangera les rôles des deux composants : le premier devient en mode secondaire et défaillant (*HSsb(1)*) et le second passe en mode primaire actif (*ON(2)*).

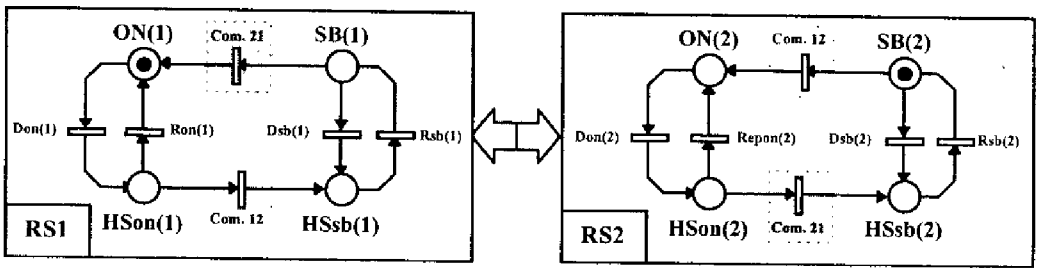


Figure VI.14: Exemple de Synchronsation Horizontale

❑

VI.4.3. Synchronisation Interne Structurée

Les principes de synchronisation internes présentés précédemment peuvent engendrer des relations très complexes entre les réseaux d'une arborescence. L'interdépendance entre chaque réseau de l'*ARS* peut ainsi être très forte.

La complexité des fonctions de synchronisation interne de chaque réseau peut être telle que la micro-évolution globale d'une *ARS* soit très difficilement intelligible. Cette complexité est notamment accrue par la non représentation graphique des synchronisations internes.

Afin de clarifier les interdépendances entre les réseaux et de structurer la micro-évolution globale d'une *ARS*, on propose d'une part d'émettre des restrictions sur le type des réseaux à synchroniser et sur la nature de la synchronisation interne et, d'autre part, de fixer une règle d'évolution globale de l'*ARS*. Cette structuration va permettre une conception plus simple des synchronisations internes entre réseaux et une compréhension plus aisée de la micro-évolution globale de l'*ARS*. On verra par ailleurs que cette forme de micro-évolution s'adapte particulièrement bien à notre domaine d'application.

Ce type de synchronisation interne sera appelé synchronisation structurée.

VI.4.3.1. Synchronisation Interne Structurée

Les contraintes à émettre sur le type des réseaux à synchroniser et sur la nature de la synchronisation interne sont regroupées dans la définition d'une *ARS* à synchronisation interne structurée.

Au préalable, on définit la propriété d'évolution unitaire que devront satisfaire les réseaux de l'*ARS*. Elle traduit le fait qu'un réseau ne doit pas pouvoir évoluer plusieurs fois consécutivement dans un même tir itéré.

Rappel

L'ensemble $T_{x,M}$ est l'ensemble des transitions d'un réseau réceptives à l'événement x pour le marquage M.

Définition VI.18 ***Evolution Unitaire***

Soit $ARS = <\Sigma_{RS}, RS0, \Gamma>$, une arborescence de *RSync* synchronisée et M un marquage atteignable quelconque : $M \in A_A(ARS; M_0)$.
Soit $RSi \in \Sigma_{RS}$ un réseau de l'arborescence.
$\forall x \in E_i \cup Sync(T_i)$ événement externe ou interne de RSi, RSi est à *évolution unitaire* ssi
$\forall t \in T_i$, $t \in T_{x,M}$, toute *SSC* bâtie sur $T_{x,M}$ et comprenant t est telle que le marquage M' global de l'*ARS* consécutif au tir de cette *SSC* ne rend aucune autre transition interne de RSi franchissable.

❑

Définition VI. 19 ***ARS à Synchronisation Interne Structurée***

Soit $ARS = <\Sigma_{RS}, RS0, \Gamma>$ une *ARS* à synchronisation horizontale et verticale.
$\forall RSi \in \Sigma_{RS}$, on a :
$T_i = T_{Ei} \cup T_{Si}$, $T_{Ei} = T_{Ei}^{P} \cup T_{Ei}^{NP}$, $T_{Si} = T_{Si}^{+} \cup T_{Si}^{-}$
$P_i = P_{Ei} \cup P_{Si}$, $P_{Si} = P_{Si}^{+} \cup P_{Si}^{-} \cup P_{Si}^{+-}$
Cette *ARS* est une *ARS* à synchronisation interne structurée ssi :

1. $\forall RSi \in \Sigma_{RS}$, RSi est une machine à états *1*-bornée à évolution unitaire
2. $\forall RSi \in \Sigma_{RS}$-{*RS0, f(ARS)*},
   si $t_i \in T_{Si}^{+}$ alors $t_i^{\bullet} \in P_{Ei} \cup P_{Si}^{-}$
3. Pour *RS0*,
   si $t \in T_{S0}^{+}$ alors $t^{\bullet} \in P_{E0}$
   si $t \in T_{E0}$ alors $t^{\bullet} \in P_{E0} \cup P_{Si}^{+}$
5. $\forall RSi \in \Sigma_{RS}$, $\forall t_i \in T_{Si}$ $M_0(RSi) \xrightarrow{t_i}$ est faux

❑

Commentaires

- La première condition impose pour tous les réseaux de l'arborescence que, d'une part, un événement interne émis par un réseau soit équivalent à son marquage et que, d'autre part, un réseau ne puisse évoluer dans deux *SSC* consécutives d'un tir itéré.
- La seconde condition concerne les réseaux non feuilles et non racine.
  Les places avales des transitions internes synchronisées sur les réseaux fils peuvent émettre des événements internes uniquement vers le réseau père.
- La troisième condition concerne le réseau racine. Seules les places avales des transitions externes (partagées ou non) peuvent émettre des événements internes vers les réseaux fils.
- La dernière condition impose qu'aucune transition interne ne soit franchissable pour la marquage initial.

Soit $\cup RSj$ un ensemble de réseaux de même niveau, les *SSC* composées sur des transitions de $\cup T_{Sj}^{+}$ sont appelés *SSC ascendantes* et celles composées sur des transitions $\cup T_{Sj}^{-}$ sont appelées *SSC descendantes*.

VI.4.3.2. Algorithme de Micro-Evolution Structurée

L'algorithme suivant décrit l'interprétation de la micro-évolution d'une *ARS* à synchronisation interne structurée. Elle est basée sur trois formes de tir itéré :

- la forme ascendante
- la forme initialement descendante
- la forme mixte

Le tir d'une transition externe (partagée ou non) peut engendrer l'émission d'un événement interne. Dans le cas contraire il y a arrêt de la micro-évolution.

Si l'événement émis est un événement interne vers le réseau père alors si une transition interne de ce réseau est franchissable elle est tirée. Elle peut émettre un événement interne uniquement vers le réseau père (condition *2.*) et le processus se renouvelle jusqu'au tir éventuel d'une transition interne du réseau racine. La micro-évolution est alors de *forme ascendante*, elle est décrite par la procédure "PropagationAscendante". Elle se termine au plus tard par le tir d'une transition interne de *RS0* (condition *3.*).

Si l'événement émis est un événement interne vers les réseaux fils alors :

- si t_i est une transition externe fusionnée, on considère les réseaux fils des deux réseaux auxquels appartient t_i.
- si t_i est une transition externe non fusionnée, on considère les réseaux fils de *RSi*.

On franchit une *SSC* sur l'ensemble de ces réseaux fils ayant une unique transition interne franchissable. Si parmi eux certains sont des réseaux feuilles, ils sont mémorisés dans une pile.

Le processus se répète tant qu'il existe des transitions internes franchissables ou que les réseaux feuilles du dernier niveau ne sont pas atteints. Ce processus est décrit dans la procédure "PropagationDescendante".

Une fois la propagation descendante terminée, la propagation ascendante est réalisée sur les réseaux préalablement empilés.

Cette micro-évolution correspond à la *forme initialement descendante*. Elle se termine au plus tard par le tir d'une transition interne de *RS0* (condition *3.*).

Si l'événement émis est un événement interne vers les réseaux fils et le réseau père alors il y a obligatoirement mémorisation du réseau père de *RSi* et la forme initialement descendante est appliquée. Comme il y a une mémorisation obligatoire du réseau père de *RSi* qui peut engendrer une propagation ascendante, il s'agit alors d'une *forme mixte* de micro-évolution. Elle se termine au plus tard par le tir d'une transition interne de *RS0* (condition *3.*).

Cet algorithme est décrit par la procédure "MicroEvolutionStructurée" appliquée à la transition externe franchissable d'un échéancier d'événements externes.

Algorithme de micro-évolution structurée

MicroEvolutionStructurée ($t_i \in T_{Ei}$ de $RSi \in \Sigma_{RSi}$)

variable P : pile de *RSync*
∪ *RSk* : liste de *RSync* ;

$P = \emptyset$; ∪ *RSk* = ∅;
Franchir t_i;

si ($t_i^\bullet \in P_{Si}^-$) alors *// Cas d'une forme ascendante*
PropagationAscendante [Γ^{-1} (*RSi*)] ;

si ($t_i^\bullet \in P_{Si}^+$) alors *// Cas d'une forme initialement descendante*
début
si ($t_i \in T_{Ei}^P$) alors *// t_i est une transition fusionnée*
début
Soit *RSj* le réseau frère de *RSi* partageant t_i;
∪ *RSk* = *RSi* ∪ *RSj* ;
fin ;
sinon ∪ *RSk* = *RSi* ;
PropagationDescendante [Γ(∪*RSk*)] ;
PropagationAscendante [Dépiler(*P*)] ;
fin ;

si ($t_i^\bullet \in P_{Si}^{+-}$) alors *// Cas d'une forme mixte*
début
Empiler [Γ^{-1} (*RSi*)] ; *// Mémorisation du réseau père*
si ($t_i \in T_{Ei}^P$) alors *// t_i est une transition fusionnée*
début
Soit *RSj* le réseau frère de *RSi* partageant t_i;
∪ *RSk* = *RSi* ∪ *RSj* ;
fin ;
sinon ∪ *RSk* = *RSi* ;
PropagationDescendante [Γ(∪*RSk*)] ;
PropagationAscendante [Dépiler(*P*)] ;
fin ;

fin .

❑

PropagationDescendante (∪*RSj*)

variables ∪*RSk*, ∪*RSl* : listes de *RSync* ;

Soit ∪*RSk* tel que [(∪*RSk* ∈ ∪*RSj*) ∧ (∀*RSk* ∃! t_k tel que $M_{RSk} \xrightarrow{t_k}$)] ;
// ∪RSk : ensemble des réseaux de ∪RSj ayant une transition interne franchissable

si (∪*RSk* ≠ ∅) alors
début
∪*RSl* = ∅ ;
Pour chaque [*RSk* ∈ *f*(*ARS*)] faire ∪*RSl* = (∪*RSl*) ∪ *RSk* ;

```
                // ∪RSl : ensemble des réseaux de ∪ RSk feuilles
                si (∪RSl ≠ ∅) alors Empiler[Γ⁻¹ (∪RSl)] ;
                // Mémorisation des réseaux pères de ∪RSl
                Franchir SSC sur {∪ tk } ;
                si [Γ(∪ RSk - ∪RSl) ≠ ∅] alors
                      PropagationDescendante[Γ(∪ RSk - ∪RSl)] ;
                // s'il existe des réseaux RSk non feuilles alors la propagation se poursuit
                // sur leur descendance
        fin ;
fin .
```

❑

```
PropagationAscendante(∪RSj)

 variable ∪RSk, ∪RSl : listes de RSync ;

 ∪RSl = Dépiler(P) ;
 // ∪RSl : ensemble de réseaux de même niveau préalablement empilés

 si [n(∪RSj) = n(∪RSl)] alors ∪RSj = (∪RSj) ∪ (∪RSl)
 // si ∪RSl et ∪RSj sont de même niveau on ajoute ∪RSl à ∪RSj
 sinon Empiler(∪RSl) ;   // sinon on rempile ∪RSl

 Soit ∪RSk tel que [(∪RSk ∈ ∪RSj) ∧ (∀RSk ∃! tk tel que M_RSk --tk-->)] ;
 // ∪RSk : ensemble des réseaux de ∪RSj ayant une transition interne franchissable

 si (∪RSk ≠ ∅) alors Franchir SSC sur {∪ tk } ;

 si (∪RSk ≠ RS0) alors PropagationAscendante[Γ⁻¹ (∪ RSk)] ;
 // si ∪RSk ne correspond pas au réseau racine alors on recommence la propagation
 // vers les réseaux pères de ∪RSk

fin .
```

❑

La structure arborescente de l'*ARS* se justifie alors pleinement car elle devient le support de la micro-évolution structurée d'une *ARS* synchronisée.

VI.4.3.3. Illustration des 3 Formes de Micro-Evolution

Les figures suivantes illustrent les trois formes de tirs itérés possibles d'après l'algorithme de micro-évolution structurée d'une *ARS* synchronisée.

La Figure VI.15 décompose les étapes d'un tir itéré ascendant et maximal. Il débute par le tir d'une transition externe dans le réseau *RS9* ① puis est suivi de *2 SSC* ascendantes composées chacune d'une seule transition ② et ③ qui atteignent la racine *RS0* de l'ARS ④. Il s'arrête alors obligatoirement.

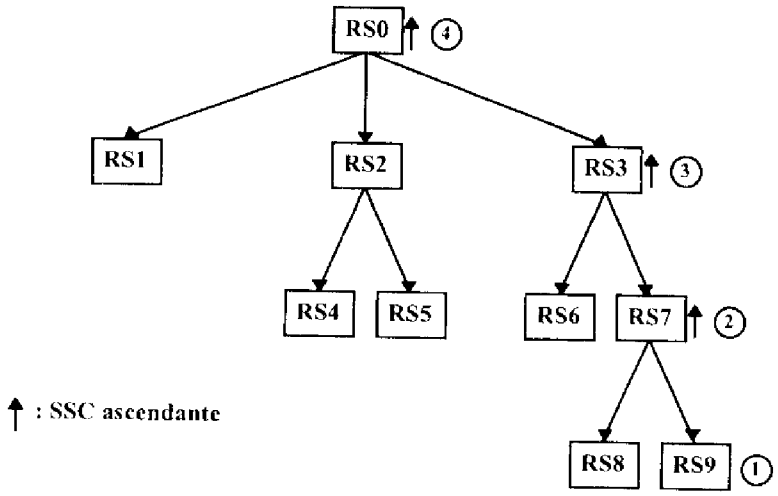


Figure VI.15: Tir Itéré Ascendant et Maximal

La Figure VI.16 décompose les étapes d'un tir itéré initialement descendant et maximal. Il débute par le tir d'une transition externe dans les réseau *RS3* ①, est suivi de *2 SSC* descendantes sur les réseaux fils ② et petits fils ③ de *RS3*. Les réseaux feuilles (*RS8* et *RS9*) sont donc atteints et la micro-évolution peut alors se poursuivre vers les ascendants de *RS8* et *RS9* : le père ④, le grand-père ⑤ et l'aïeul racine ⑥.

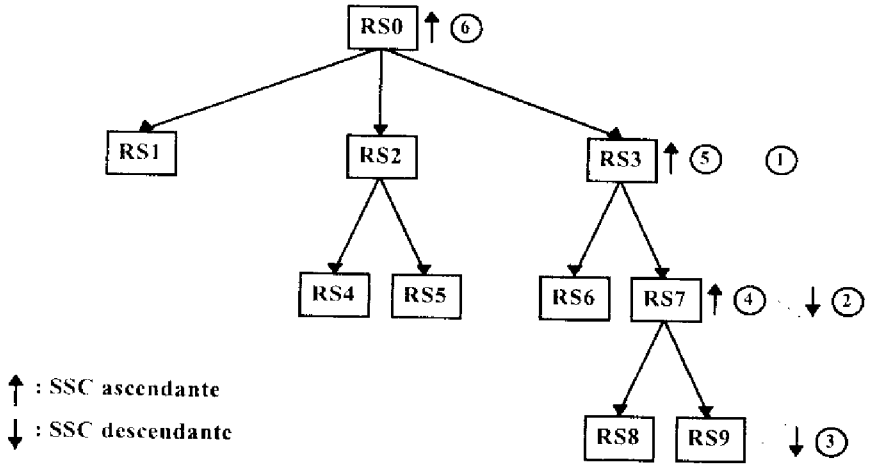


Figure VI.16: Tir Itéré Initialement Descendant et Maximal

La figure suivante décompose les étapes d'un tir itéré mixte et maximal pour une transition externe fusionnée. Il débute par le tir d'une transition externe commune à *RS2* et *RS3* ① se poursuit par le tir d'une *SSC* descendante sur les réseaux fils (*RS4, RS5, RS6* et *RS7*) et petits fils (*RS8* et *RS9*) de *RS2* et *RS3* : ② et ③. Les réseaux feuilles (*RS8* et *RS9*) sont donc atteints et la micro-évolution peut alors se poursuivre vers le père *RS7* de *RS8* et *RS9* ④, elle se prolonge ensuite sur le père de *RS7* et ceux des réseaux feuilles déjà explorés du même niveau que *RS7* ⑤ et se termine par une *SSC* sur l'aïeul commun : le réseau racine ⑥.

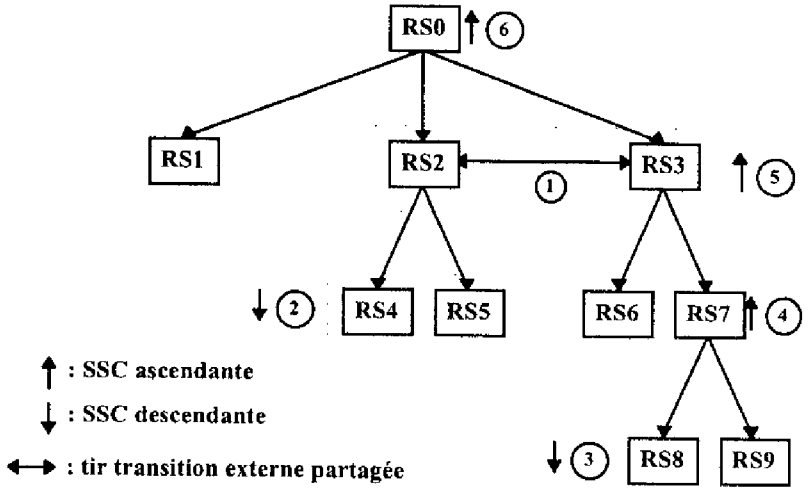


Figure VI.17: Tir Itéré Mixte et Maximal pour une Transition Externe Fusionnée

VI.4.3.4. Exemple

Reprenons l'exemple de la Figure VI. 12, l'ARS décrite est une ARS à synchronisation interne structurée. En effet :

- Tous les réseaux sont des machines à états 1-bornées.
- Chaque réseau est à évolution unitaire.
- Chaque transition interne, des réseaux non feuilles et non-racine, synchronisée sur des réseaux fils peut engendrer un événement interne vers le père.
- Une transition interne du réseau racine ne peut engendrer aucun événement interne vers les réseaux fils.
- Le marquage initial est tel qu'aucune transition interne n'est franchissable.

La micro-évolution structurée pour son marquage initial de cette ARS pourra prendre deux formes :

- La forme ascendante : elle correspond à la propagation de défaillances.
- La forme initialement descendante : elle correspond à la propagation de maintenance globale.

Ces notions de propagation de défaillances et propagation de maintenances seront largement reprises pour caractériser l'évolution des *arbres de défaillances dynamiques* au chapitre suivant.

VI.5. Conclusion

Ce chapitre a permis de définir le modèle que nous utiliserons par la suite pour bâtir les arbres de défaillances dynamiques : l'arborescence de *RSync* à synchronisation interne structurée.
Ce modèle autorise deux formes de synchronisation interne entre les réseaux de l'arborescence: la synchronisation horizontale et la synchronisation verticale. Ces deux types de synchronisation permettent de caractériser soit une évolution simultanée de deux réseaux, soit une évolution d'un ou plusieurs réseaux consécutive à celle d'un ou plusieurs autres réseaux de niveau adjacent.
Des contraintes sur la nature des réseaux et de leurs synchronisations ont été émises afin de structurer la micro-évolution globale de l'arborescence décrite par l'algorithme de micro-évolution. Ce dernier offre trois formes de micro-évolution (forme ascendante, initialement descendante et mixte) qui permettront de caractériser les processus de défaillance, maintenance externe et maintenance interne définis par la suite.

Chapitre VII

Arbres de Défaillances Dynamiques

VII.1. Introduction

Les *Arbres de Défaillances Dynamiques* [Ereau 95] sont des arborescences de réseaux de Petri à synchronisation structurée qui modélisent les liens de dépendance entre les éléments d'un système du point de vue des défaillances bien sûr mais également des maintenances, fin de vie et initialisations. Ces modèles peuvent être complexes graphiquement mais ils sont totalement transparents aux utilisateurs car automatiquement générés à partir des *Arbres de Défaillances Paramétrés*.

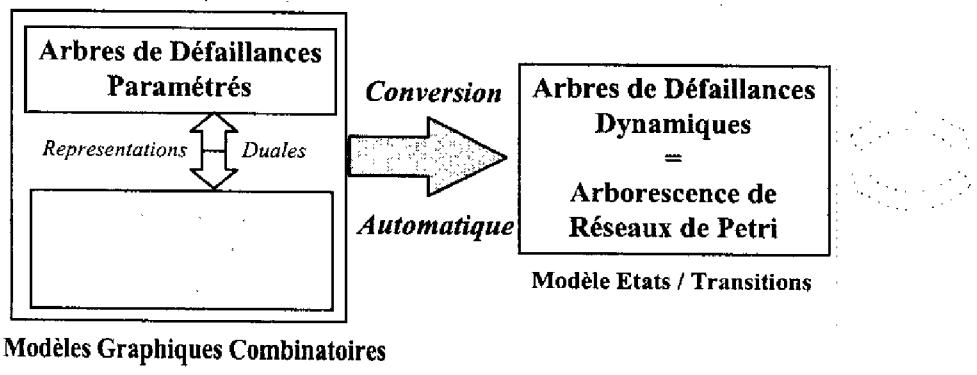


Figure VII.1 : Génération des *Arbres de Défaillances Dynamiques*

La génération automatique *d'Arbres de Défaillances Dynamiques* permet donc le passage d'un modèle combinatoire augmenté de paramètres à un modèle Etats-Transitions. Ce dernier est simulable ou analysable par calcul Markovien mais peut également être couplé avec des *modèles ressources* pour offrir un modèle global. C'est ce que rappelle la Figure VII.1.

Ce dernier Chapitre est dédié à la mise en oeuvre concrète de l'aide à la modélisation des *modèles structurels* qui repose sur cette génération d'*Arbres de Défaillances*.

La section *2* décrit le modèle combinatoire initial à partir desquels sont générés les *Arbres de Défaillances Dynamiques (ADD)* : les *Arbres de Défaillances Paramétrés (ADP)*. Il s'agit d'une extension des Arbres de Défaillances, bien connus des fiabilistes et très simples d'emploi. De façon analogue, on aurait pu définir comme modèle combinatoire initial des *Blocs Diagrammes de Fiabilité Paramétrés*.

La section *3* présente les principes de génération des *ADD* à partir des *ADP* ainsi que leurs formes d'évolution. Ces dernières sont directement issues des *3* formes de micro-évolution structurées présentées au chapitre précédent.

La section *4* illustre ce processus d'aide à la construction d'un *modèle structurel* en reprenant l'exemple d'une constellation de satellites. C'est l'occasion de présenter brièvement l'outil informatique que nous avons développé proposant l'édition d'*ADP* [Lahorgues 95] et la traduction d'*ADP* en *ADD* [Raspaud 96].

Enfin, la dernière section positionne cette démarche par rapport à d'autres développées parallèlement et propose des directions pour l'extension des *ADD* et leur couplage avec des *modèles ressources*.

VII.2. Arbres de Défaillances Paramétrés (ADP)

VII.2.1. Principe Général

Définition VII. 1 ***Arbres de Défaillances Paramétrés***

Un *Arbre de Défaillances Paramétré (ADP)* est:

- un Arbre de Défaillances classique : portes ET et OU uniquement,
- sans événements répétés,
- dont les événements indésirables de base sont limités aux défaillances de composants du système modélisé,
- dont la structure arborescente est directement liée à l'architecture matérielle,
- pour lequel, à chaque sommet, on associe un certain nombre de paramètres caractérisant maintenance, initialisation et durée de vie.

❑

Notations

On appelle:

- ***Composant*** : un constituant « atomique » dans la description d'un système,
- ***Macro-composant*** : un constituant non atomique c'est à dire ayant une descendance hiérarchique,
- ***Elément*** : un *composant* ou *macro-composant*.

D'après cette définition, l'Arbre de Défaillances sous-jacent à un *ADP* décrit les conditions logiques de propagation de défaillances vers les macro-composants et jusqu'au système lui-même. Par exemple, l'*ADP* décrivant une constellation de satellites composée de *2* plans d'orbites ayant chacun *2* satellites pourrait avoir comme arbre sous-jacent celui de la Figure VII.2. Les satellites sont les composants qui peuvent défaillir, les orbites sont les macro-composants tandis que la constellation est la racine de l'arborescence représentant le niveau système.

Par abus, les noeuds de l'arborescence sont assimilés aux éléments matériels qui peuvent devenir défaillants plutôt qu'aux événements « défaillance » de ces éléments. Si bien que l'arborescence d'événements devient une arborescence d'éléments mais dont la structure traduit toujours la propagation de défaillances.

Les Arbres de Défaillances Paramétrés représentent toujours graphiquement les propagations de défaillance mais permettent également d'avoir accès, grâce aux paramètres de sommet, aux informations nécessaires pour caractériser les processus de maintenance, et éventuellement de durée de vie et d'initialisation.

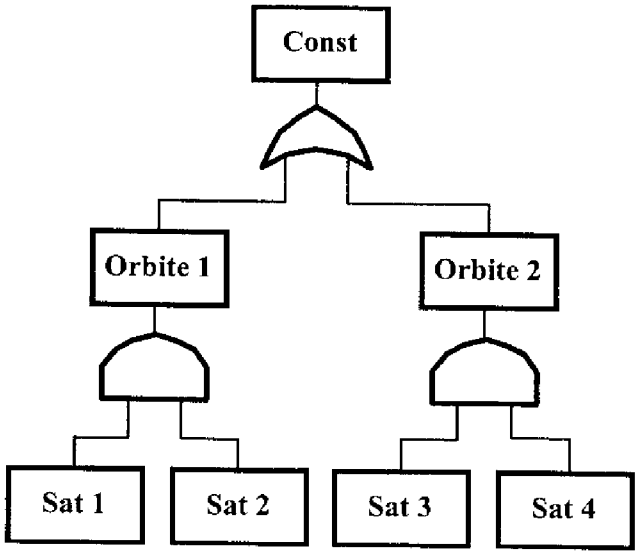


Figure VII.2 : Exemple d'Arbre de Défaillances sous-jacent à un *ADP*

VII.2.2. Types de Sommet

Les sommets d'un *ADP* sont caractérisés par la forme de maintenance qui peut s'appliquer à l'élément du système qu'ils représentent. Dans le cadre de notre étude deux types de maintenance ont été définis : la maintenance externe et la maintenance interne.

Maintenance Externe et Maintenance Interne

La *maintenance externe* est un processus extérieur à la structure système dont l'objectif est de réparer ou remplacer un élément défaillant, l'élément ainsi maintenu se retrouve alors comme neuf.

La *maintenance interne* pallie la défaillance d'un élément par simple commutation sur un élément de secours, pouvant remplir la même fonction que l'élément défaillant et faisant déjà partie intégrante du système.

Conventions

La *maintenance externe* appliquée à un sommet d'un *ADP* signifie que c'est ce sommet et tous ses descendants dans l'arborescence qui peuvent être réparés ou remplacés.
La *maintenance interne* appliquée à un sommet d'un *ADP* signifie que ce sommet représente un macro-élément dont les fils sont soit actifs, soit en attente. Si l'un des fils actifs est défaillant alors, il sera remplacé par l'un des fils en attente et non défaillant. Les rôles de ces deux éléments fils sont alors échangés: l'un devient défaillant et en attente tandis que l'autre devient actif et en bon fonctionnement.

Caractérisation d'un Type de Sommet

Le type d'un sommet d'un *Arbre de Défaillances Paramétré* est donc défini à partir:

- du type de maintenance qui s'applique à son niveau : aucune, interne et/ou externe,
- de sa position dans l'arborescence : feuille ou non,
- de son état initial : actif ou en attente.

Au total *16* types de sommet sont donc envisageables.

Hypothèse Simplificatrice

Dans le cadre de notre étude, afin de limiter l'explosion combinatoire des possibilités de maintenance des éléments d'un système, on a supposé que tout élément peut être soumis, directement, par sa descendance ou son ascendance, au plus une fois à chaque type de maintenance.

Types Licites de Sommet et Symboles Associés

La convention choisie sur la *maintenance interne* et l'hypothèse qui précède limitent donc à *10* le nombre de types de sommet autorisés. La Figure VII.3, extraite de l'éditeur d'*ADP* développé, les présente ainsi que la symbolique retenue.

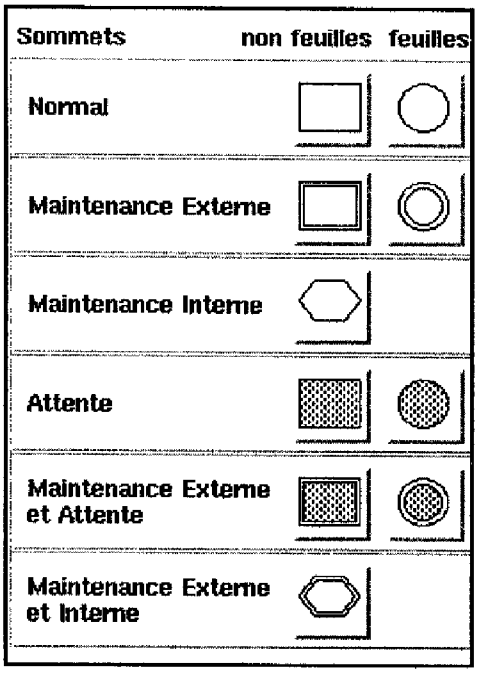


Figure VII.3 : Types de Sommet proposés par l'Editeur d'*ADP*

VII.2.3. Paramètres de Sommet

Les différents types de sommet définis ci-avant conjugués à la structure arborescente des *ADP* permettent de décrire la logique de maintenance associée à chaque élément d'un système. Pour quantifier ces processus ainsi que les défaillances naturellement prises en compte, on attribue à chaque sommet un certain nombre de paramètres. Ces paramètres

peuvent varier selon le type de sommet considéré, mais dans tous les cas, on retrouve des paramètres obligatoires et des paramètres optionnels.

Paramètres Obligatoires

- Pour tout sommet feuille, le paramètre « *Défaillance* » doit être renseigné. Il s'agit du type de loi de défaillance (exponentielle, Erlanguienne, ...) de l'élément et des attributs qui vont avec. Si un élément peut être en mode attente (c'est le cas si l'un de ses ascendants est soumis à la *maintenance interne*) alors le paramètre « *Défaillance en attente* » doit également être renseigné.
- Pour tout sommet auquel s'applique la *maintenance externe* un type de processus doit être sélectionné. Si l'option « *taux* » est choisie, à l'instar du paramètre « *Défaillance* », le paramètre « *Réparation* » doit être renseigné. La maintenance est alors vue comme un processus atomique caractérisé par une unique transition entre état. Si l'option « *ressource* » est sélectionnée, alors le *modèle ressource* doit être désigné. Dans ce dernier cas, la maintenance externe correspond à un processus plus complexe décrit par un réseau de Petri.
- Pour tout sommet auquel s'applique la *maintenance interne*, le paramètre « *Commutation* » doit être renseigné. Ce paramètre décrit la loi (et ses attributs) qui caractérise la commutation entre deux éléments fils.

Paramètres Optionnels

Les paramètres optionnels concernent exclusivement la prise en compte de l'initialisation globale du système et de la durée de vie de ses éléments.

- Au sommet racine d'un *ADP* on peut associer le paramètre « *Initialisation* » qui en fait désigne le fichier d'un *modèle ressource* dont le rôle est de décrire l'initialisation des éléments du système.
- A chaque sommet, on peut associer le paramètre « *Durée de vie* ». Ce paramètre décrit la loi (et ses attributs) qui caractérise la durée de vie de l'élément. Dans le cas où ce paramètre est positionné il ne peut alors être pris en compte pour aucun élément de sa descendance.

VII.2.4. Exemple

La Figure VII.4 propose un exemple d'*ADP* pour lequel seuls les types de sommet sont considérés. L'exemple final du Chapitre décrit également l'utilisation des différents paramètres.

La numérotation des sommets est réalisée en largeur d'abord. Le sommet *0* racine de l'arborescence représente un élément composé de *3* fils (*1, 2, 3*). La défaillance de l'un d'entre eux entraîne celle du système (choix de la porte *OU*). Aucune maintenance n'est directement appliquée à son niveau.

Le sommet *1* représente un *macro-composant* auquel s'applique la *maintenance interne*. C'est à dire que ses fils (sommets feuilles) décrivent des *composants* en redondance passive. Initialement le composant *4* est actif tandis que le composant *5* est en attente.

Le sommet *3* correspond à un macro-composant auquel s'applique la maintenance externe. Ses fils feuille sont en redondance active (porte *ET*).

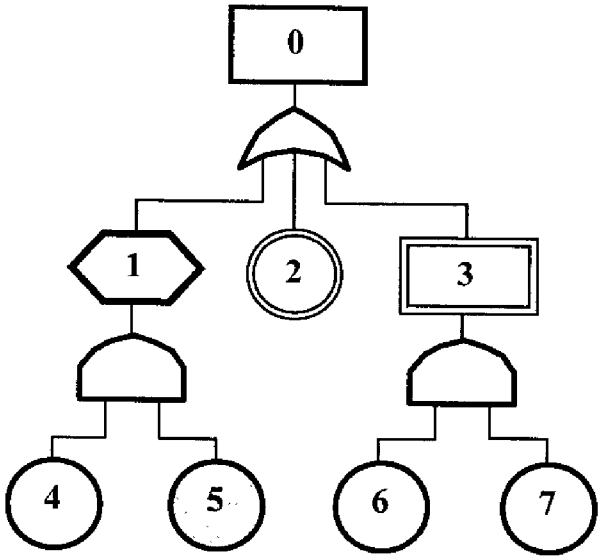


Figure VII.4 : Un Exemple d'Arbre de Défaillance Paramétré

VII.3. Principe des ADD

Nous venons de présenter les *Arbres de Défaillances Paramétrés* qui permettent la saisie aisée des liens de dépendance entre les éléments d'un système. Nous présentons maintenant la construction automatique des *Arbres de Défaillances Dynamiques* à partir des *ADP* ainsi que les principes d'évolution de ce modèle. Pour une description plus exhaustive se reporter à [Ereau 95(2)].

VII.3.1. Génération Automatique : Synchronisation Structurée d'une Arborescence de Réseaux Prédéfinis

Comme on a pu le souligner, si les *Arbres de Défaillances Paramétrés* contiennent l'information utile pour décrire les propagations de maintenance, de fin de vie et d'initialisation, ils ne sont pas utilisables en tant que tels. Il est donc nécessaire de convertir ce formalisme simple d'emploi en un modèle exploitable par calcul analytique ou par simulation. Ce modèle ou *Arbre de Défaillances Dynamique* est en fait un modèle Etats-Transitions décrit par une arborescence de réseaux à synchronisation interne structurée définie au Chapitre VI.

La génération automatique d'*ADD* à partir d'un *ADP* a donc pour but de construire cette arborescence de réseaux et de réaliser les synchronisations internes (verticales et horizontales). On peut alors s'appuyer sur les règles d'évolution proposées pour caractériser la propagation des défaillances bien sûr, mais également celle des maintenances, fins de vie et initialisations.

La construction automatique de *l'ADD* est réalisée à partir de l'analyse de l'*ADP* en deux passes successives.

Lors de la première passe, pour chaque sommet X de l'*ADP*, on construit un réseau prédéfini *R(X)* dont la structure dépend du type de X et de sa position dans l'arborescence. Ce réseau décrit les états et transitions possibles de l'élément associé à X. C'est, par hypothèse, une machine à états *1*-bornée dont toute place marquée donne directement l'état de X.
La synchronisation *externe* du réseau est réalisée. Toute transition de *R(X)* correspondant à un événement externe propre à X (du niveau de *X)* devient une transition externe dont la temporisation est décidée à partir de la loi (et ses attributs) de l'événement lue sur les paramètres de X.

Chaque réseau *RS(X)* ayant été préalablement construit et l'arborescence de l'*ADP* ayant été visitée une fois, la seconde passe permet de réaliser la synchronisation interne. Cette dernière décrit les interactions entre les éléments d'une même branche (synchronisation verticale) et entre les éléments frères (synchronisation horizontale). Ces synchronisations mettent en oeuvre la structure permettant de décrire les propagations de défaillance, de maintenance, de fin de vie et d'initialisation.

Notations

- Comme au Chapitre précédent la fonction Γ est la fonction successeurs de l'arborescence tandis que son inverse Γ^{-1} est la fonction prédécesseur (un seul prédécesseur par définition).

- A chaque élément X le réseau associé est noté $R(X)$ de même que chacune des places P de ce réseau est notée $P(X)$. $P(X)$ peut également caractériser l'événement interne « place P du réseau $R(X)$ marquée ».
- $\prod\{P[\Gamma(X)]\}$ est l'événement interne associé à une transition interne de $R(X)$. Il signifie que l'ensemble des places de nom P des réseaux fils de $R(X)$ doivent être marquée.
- $\sum\{P[\Gamma(X)]\}$ est l'événement interne associé à une transition interne de $R(X)$. Il signifie que l'une au moins des places de nom P des réseaux fils de $R(X)$ doit être marquées.
- $P[\Gamma^{-1}(X)]$ est l'événement interne associé à une transition interne de $R(X)$. Il signifie que la place de nom P du réseau père doit être marquée.

La Figure VII.5 présente le modèle *ADD* généré à partir de l'Arbre de Défaillance Paramétré pris en exemple au paragraphe VII.2.4.

Chaque réseau présente au moins les places de type « *ON* » et « *HS* ». Les transitions « $Déf_i$ », présentes pour les réseaux feuilles uniquement, sont des transitions externes qui décrivent la défaillance d'un élément. Les transitions entre les places de type « *ON* » et « *HS* » pour les réseaux non-feuilles sont des transitions internes synchronisées sur les places des réseaux fils. Ceci pour décrire la propagation des défaillance au sein de l'arborescence.

Les transitions « $Maint_i$ » sont des transitions externes qui traduisent la *maintenance externe* d'un élément. La transition interne entre *HS(0)* et *ON(0)* du réseau racine est synchronisée sur les places des réseaux fils car c'est d'eux (directement ou non) que viennent les possibilités de maintenance. Les transitions des réseaux feuilles *R(6)* et *R(7)* entre les places de type « *HS* » et « *ON* » sont cette fois synchronisées sur la place « *OK* » du réseau père car c'est à son niveau qu'est initié la *maintenance externe*.

Le réseau *R(1)* représente un élément auquel s'applique la *maintenance interne* il est donc légitime que sa transition interne entre « *HS(1)* » et « *ON(1)* » soit synchronisée sur ses réseaux fils dont les éléments sont en redondance passive. Cette dernière est mise en oeuvre grâce au partage de la transition externe (ou synchronisation horizontale) « $C_{4/5}$ » entre les réseaux *R(4)* et *R(5)*. Ces derniers ont d'ailleurs plus de places donc plus d'états possibles, ceci afin de prendre en compte les possibilités de mode actif ou attente.

La description plus précise des modes de propagation de défaillances et maintenances externes et/ou internes fait l'objet de la section suivante consacrée à l'évolution d'un *Arbre de Défaillances Dynamique*.

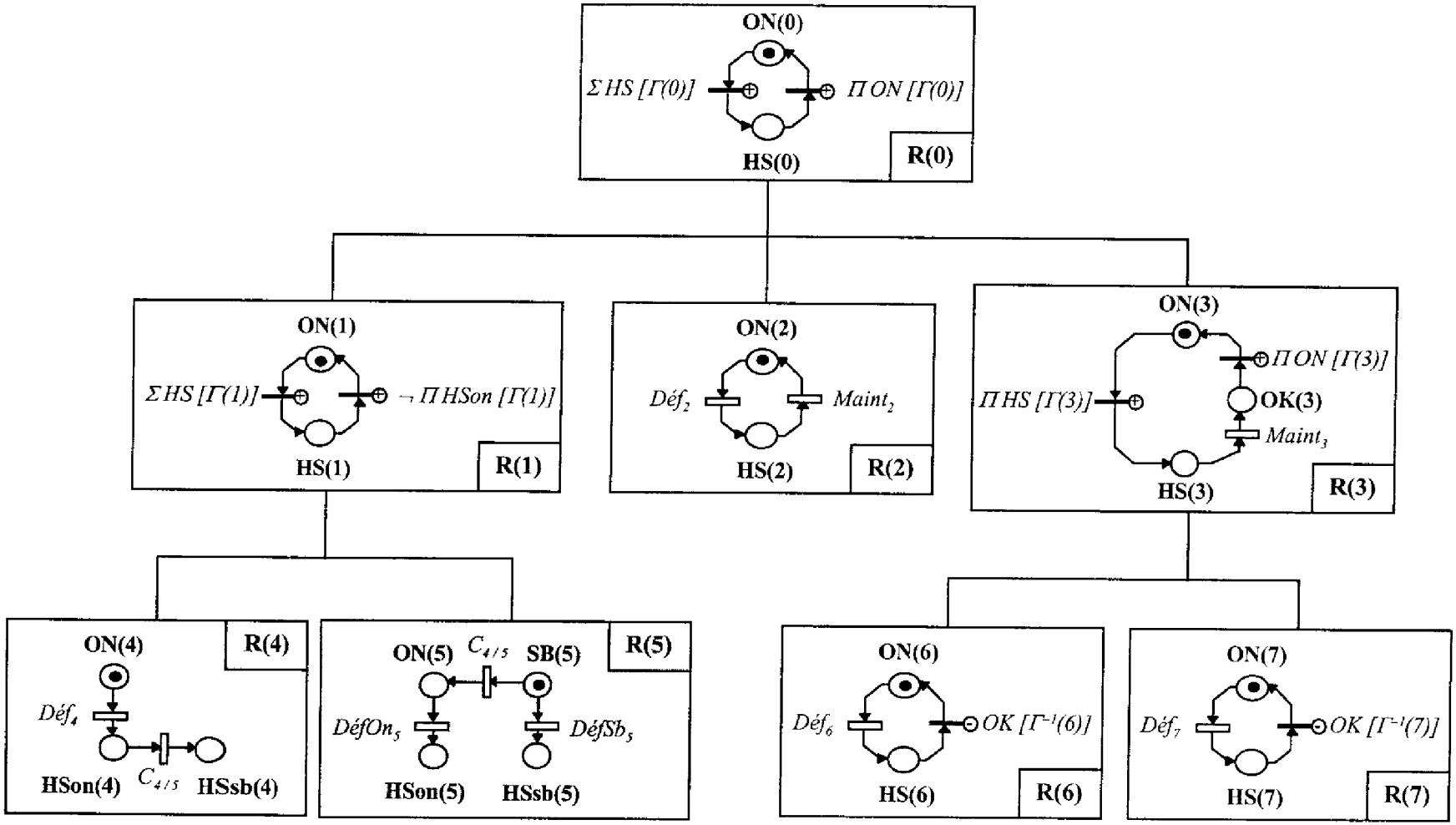


Figure VII.5 : Arborescence Générée à partir de l'*ADP* de la Figure VII.4

VII.3.2. Evolution : Algorithme de Micro-Evolution Structurée

L'arborescence de réseaux de Petri générée est par construction à synchronisation interne structurée. Son évolution va donc être dictée par l'algorithme de micro-évolution structurée présenté au Chapitre VI.

Cinq types différents d'événements externes (défaillance, maintenance externe, maintenance interne, fin de vie et initialisation) peuvent être pris en compte au niveau de chaque élément. Ces événements externes peuvent être propagés au sein de l'arborescence. Dans ce qui suit, on va voir comment les *3* formes de micro-évolution structurée permettent de décrire ces propagations.

VII.3.2.1. Séquences de Propagation de Défaillances : Tirs Itérés Ascendants

Par construction, seules les feuilles d'un *ADP* peuvent et doivent avoir renseigné le paramètre « *Défaillance* ». En conséquence, seuls les réseaux feuilles de l'*ADD* ont une transition externe correspondant à la défaillance propre de l'élément correspondant. Un tel événement ne peut se propager que vers l'ascendance et au pire jusqu'à la racine de l'arborescence indiquant en cela que la défaillance du composant a provoqué la défaillance du système entier.

On reconnaît là la structure d'un tir itéré ascendant, une des trois formes de la micro-évolution structurée. Dans le contexte d'un *Arbre de Défaillances Dynamique* ce tir itéré est appelé *Séquence de Propagation de Défaillances (SPD).*

La Figure VII.6 décrit une *SPD* initiée par la défaillance du composant *6*.

- ① : La transition externe « $Déf_6$ » est franchie. L'événement interne *HS(6)* n'est émis que vers le réseau père *R(3)*.
- ② : L'événement $\prod\{HS[\Gamma(3)]\}$ devient vrai pour le réseau *R(3)*. La transition associée est donc franchissable et son tir représente la défaillance de l'élément *3*. L'événement interne *HS(3)* n'est émis que vers le réseau père *R(0)*.
- ③ : L'événement $\sum\{HS[\Gamma(0)]\}$ devient vrai pour le réseau *R(0)*. Il se retrouve dans l'état *HS* modélisant ainsi la défaillance du système.

VII.3.2.2. Séquences de Propagation de Maintenances Externes : Tir Itéré Initialement Descendant

Comme on l'a vu (cf. §VII.2.2) la maintenance externe peut s'appliquer à un élément *X* quel que soit son niveau dans l'arborescence de l'*ADP*. Cette maintenance est décrite dans l'*ADD* par le tir d'une transition externe non partagée du réseau *R(X)*. C'est le début d'un tir itéré initialement descendant se propageant à chacun des éléments feuilles de la descendance de *R(X)* pour remonter et gagner éventuellement son ascendance.

Ce tir itéré dans le contexte des *ADD* est appelé *Séquence de Propagation de Maintenance Externe (SPME).*

La Figure VII.7 décrit une *SPME* initiée par la maintenance externe de l'élément *3*.

- ① : La transition externe $Maint_3$ est franchie. L'événement interne *OK(3)* est émis uniquement vers les réseaux fils de *R(3)*.
- ② : L'événement $OK(3) = OK\left[\Gamma^{-1}(6)\right] = OK\left[\Gamma^{-1}(7)\right]$ devient vrai. Une *SSC* sur les réseaux *R(6)* et *R(7)* est donc franchie menant ces réseaux dans l'état *ON*.
- ③ : L'événement $\prod\left\{ON\left[\Gamma(3)\right]\right\}$ devient vrai pour le réseau *R(3)*. Ce dernier revient donc dans l'état *ON*.
- ④ : L'évenement $\prod\left\{ON\left[\Gamma(0)\right]\right\}$ devient vrai pour le réseau *R(0)*. Il revient dans l'état *ON* représentant ainsi le retour du système dans un état de bon fonctionnement.

VII.3.2.3. Séquence de Propagation de Maintenances Internes : Tir Itéré Mixte

Une maintenance interne ne peut s'appliquer qu'au niveau d'un macro-élément *X*. La commutation entre deux éléments fils de *X* ($\{X_1, X_2\} \in \Gamma(X)$) est représentée par une transition externe partagée entre les réseaux $R(X_1)$ et $R(X_2)$. C'est le début d'un tir itéré initialement mixte se propageant d'abord vers les feuilles de *X* puis directement vers son ascendance.

Ce tir itéré dans le contexte des *ADD* est appelé *Séquence de Propagation de Maintenance Interne (SPMI).*

La Figure VII.8 décrit une *SPMI* initiée par la commutation entre les éléments *4* et *5*.

- ① : La transition externe partagée $C_{4/5}$ est franchie menant le réseau *R(4)* dans l'état *HSsb* et le réseau *R(5)* dans l'état *ON*.
- ② : L'événement $\neg\prod\left\{HSon\left[\Gamma(1)\right]\right\}$ devient vrai. Le réseau *R(1)* revient dans l'état *ON*.
- ③ : L'évenement $\prod\left\{ON\left[\Gamma(0)\right]\right\}$ devient vrai pour le réseau *R(0)*. Il revient dans l'état *ON* représentant ainsi le retour du système dans un état de bon fonctionnement.

VII.3.2.4. Séquences de Propagation de Fin de Vie et d'Initialisation

La fin de vie des éléments peut être vue comme un cas particulier de défaillance de ces éléments, la défaillance n'étant pas dans ce cas limitées aux éléments feuilles. *Une Séquence de Propagation de Fin de vie (SPF)* peut alors aussi être décrite par un tir itéré ascendant.

De même, l'initialisation d'un système est équivalente à une maintenance externe. Une *Séquence de Propagation d'Initialisation (SPI)* peut donc être décrite par une *SPME* donc par un tir itéré initialement descendant.

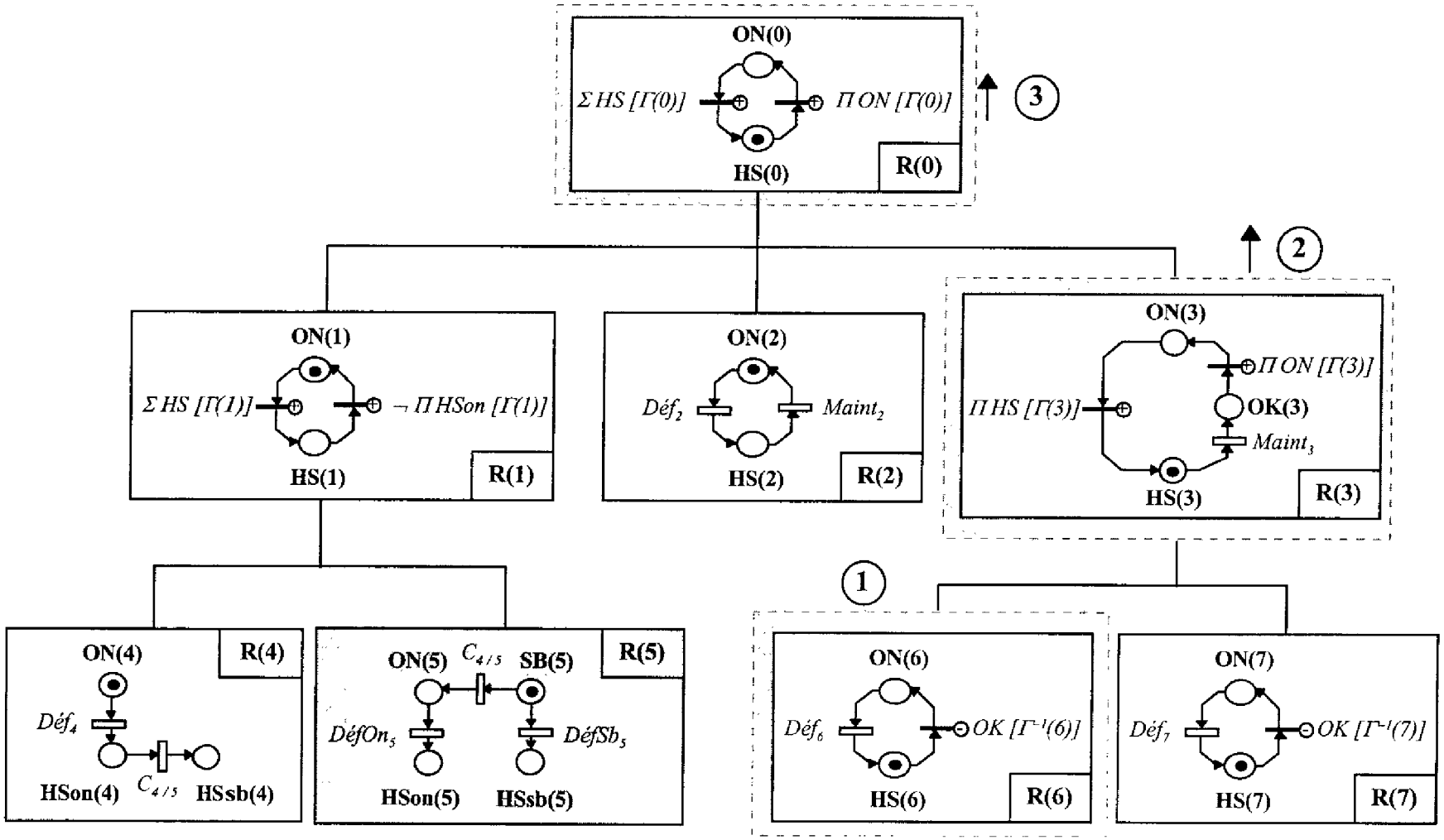


Figure VII.6 : Exemple de *Séquence de Propagation de Défaillances*

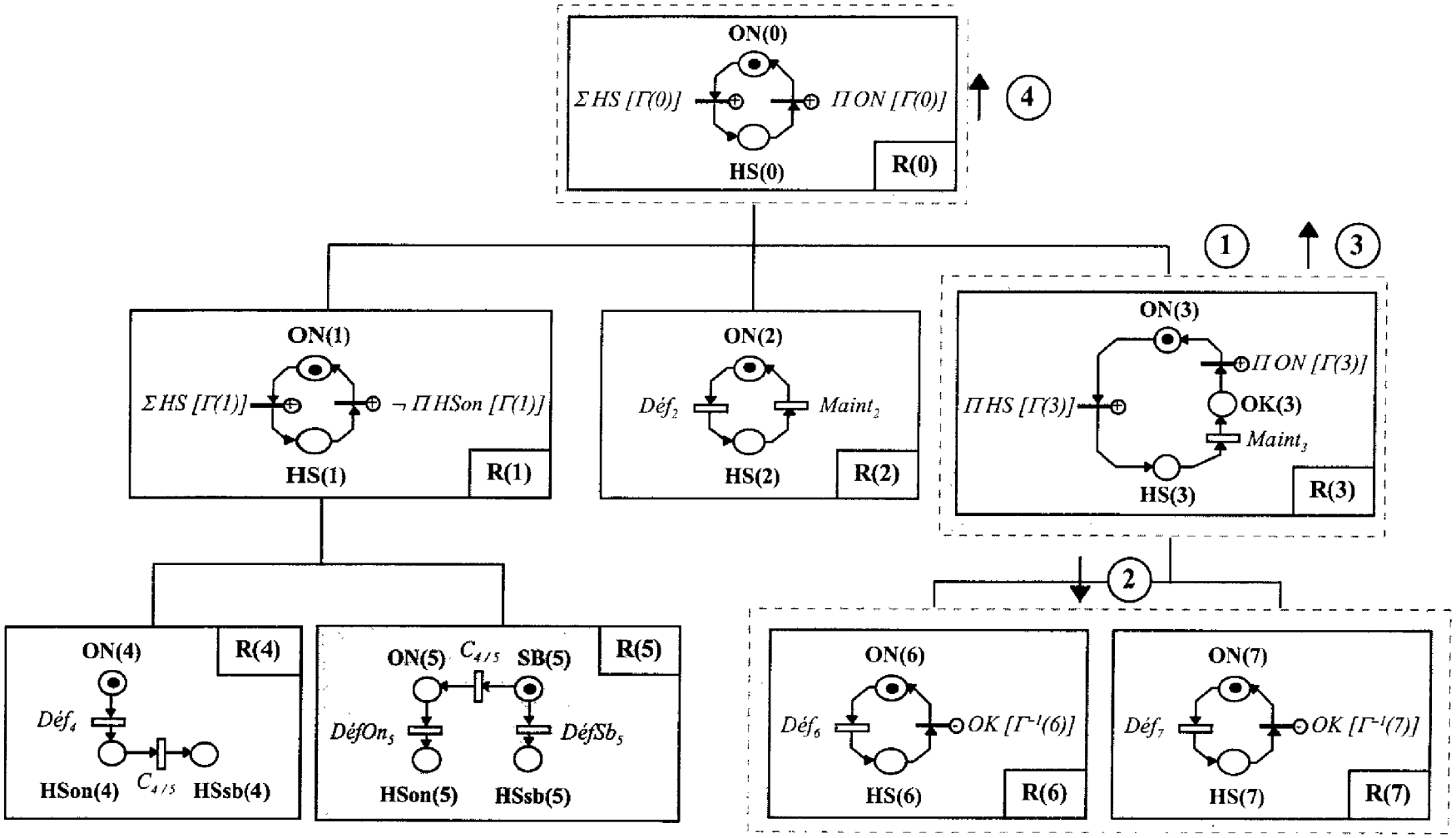


Figure VII.7 : Exemple de *Séquence de Propagation de Maintenances Externes*

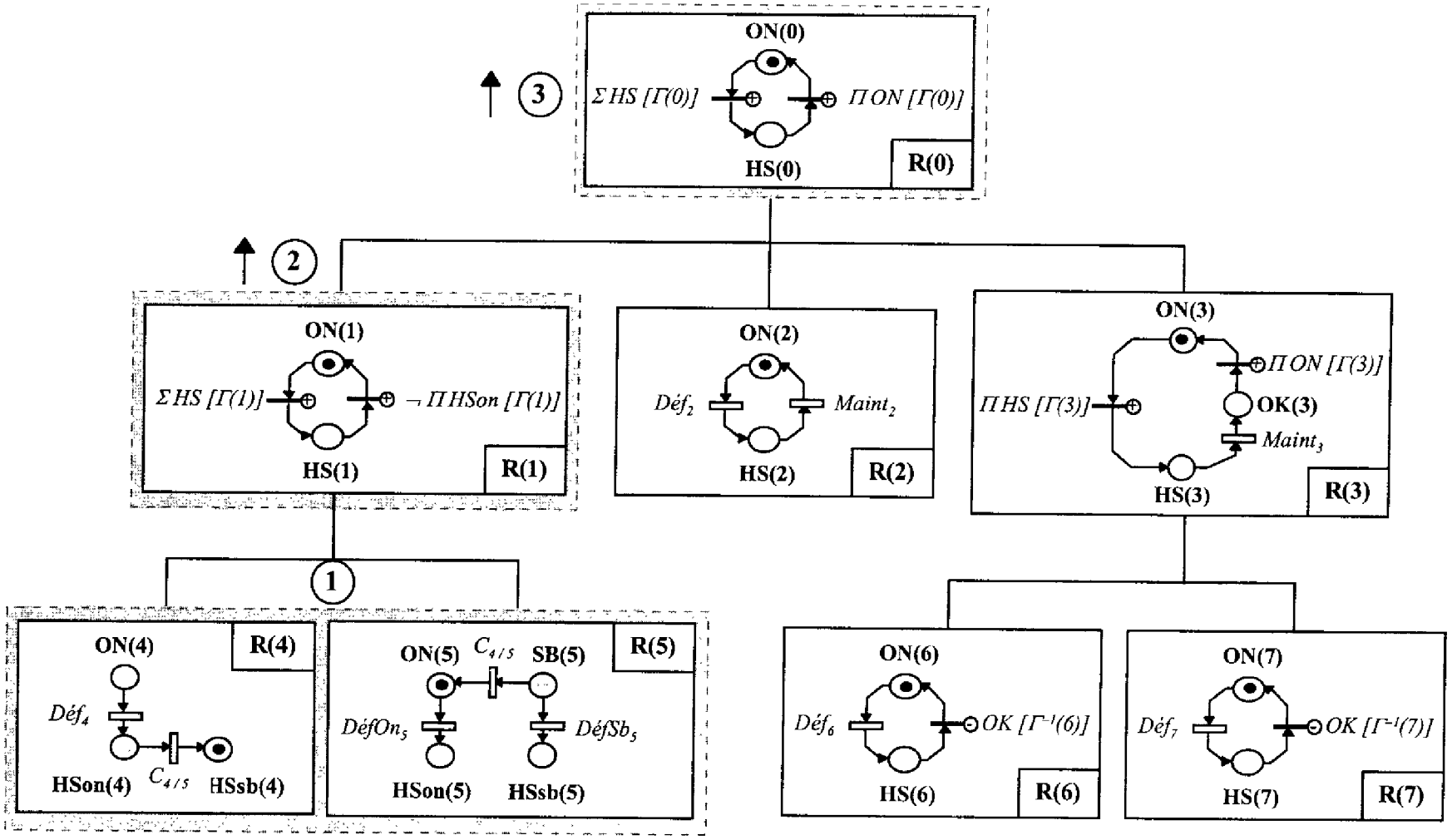


Figure VII.8 : Exemple de *Séquence de Propagation de Maintenances Internes*

VII.4. Exemple d'Application

Afin d'illustrer la démarche d'aide à la modélisation proposée nous reprenons le problème de la modélisation d'une constellation de satellites. Notre but est d'obtenir le modèle structurel décrivant les liens de dépendances entre les éléments du segment spatial. C'est l'occasion de présenter l'outil informatique d'édition *d'Arbres de Défaillances Paramétrés* et de génération Automatique d'*Arbres de Défaillances Dynamiques* qui a été développé au CNES. Cet outil n'est qu'un prototype qui nous a permis d'illustrer et de valider les concepts développés.

On suppose que la constellation est composée de *3* plans d'orbite. Sur chaque plan d'orbite, *3* satellites sont en redondance passive *2* parmi *3*. Tout satellite est composé d'une plateforme et d'une charge utile. Si l'un des satellites est défaillant alors il est remplacé (maintenance externe).

VII.4.1. Brève Description de l'Editeur

La Figure VII.9 est une capture d'écran de l'éditeur d'*ADP*. Ce dernier fonctionne sous station UNIX-MOTIF et a été codé en TCL-TK [Ousterhout 95]. On retrouve des éléments traditionnels d'une interface homme machine: la barre de menus, la palette d'icônes et la zone de travail.

La barre de menus présente, outre les menus classiques de lecture/sauvegarde et d'édition, les menus plus dédiés à notre application d'Affichage et de Traitement.
Le menu *Affichage* offre la possibilité de visualiser un certain nombre de variables et de réorganiser l'arborescence courante pour une meilleure visibilité.
Le menu *Traitement* permet de vérifier l'arborescence (de sa structure et des règles propres aux *ADP*) de lancer la génération automatique d'*ADD* à partir de l'*ADP* courant.

La palette d'icônes est divisée en *3* parties : les icônes permettant l'édition des différents types de sommets d'un *ADP*, les icônes permettant l'édition des portes et des liens, et une zone de modification de l'affichage.

VII.4.2. Edition de l'*ADP* de la Constellation

L'Arborescence présentée par la Figure VII.9 correspond au modèle de la constellation proposée. On retrouve la structure hiérarchique matérielle. Le premier niveau correspond au système, le second à chaque plan d'orbite, le troisième aux satellites et le dernier aux composants élémentaires considérés c'est à dire les charges utiles et les plateformes des satellites.

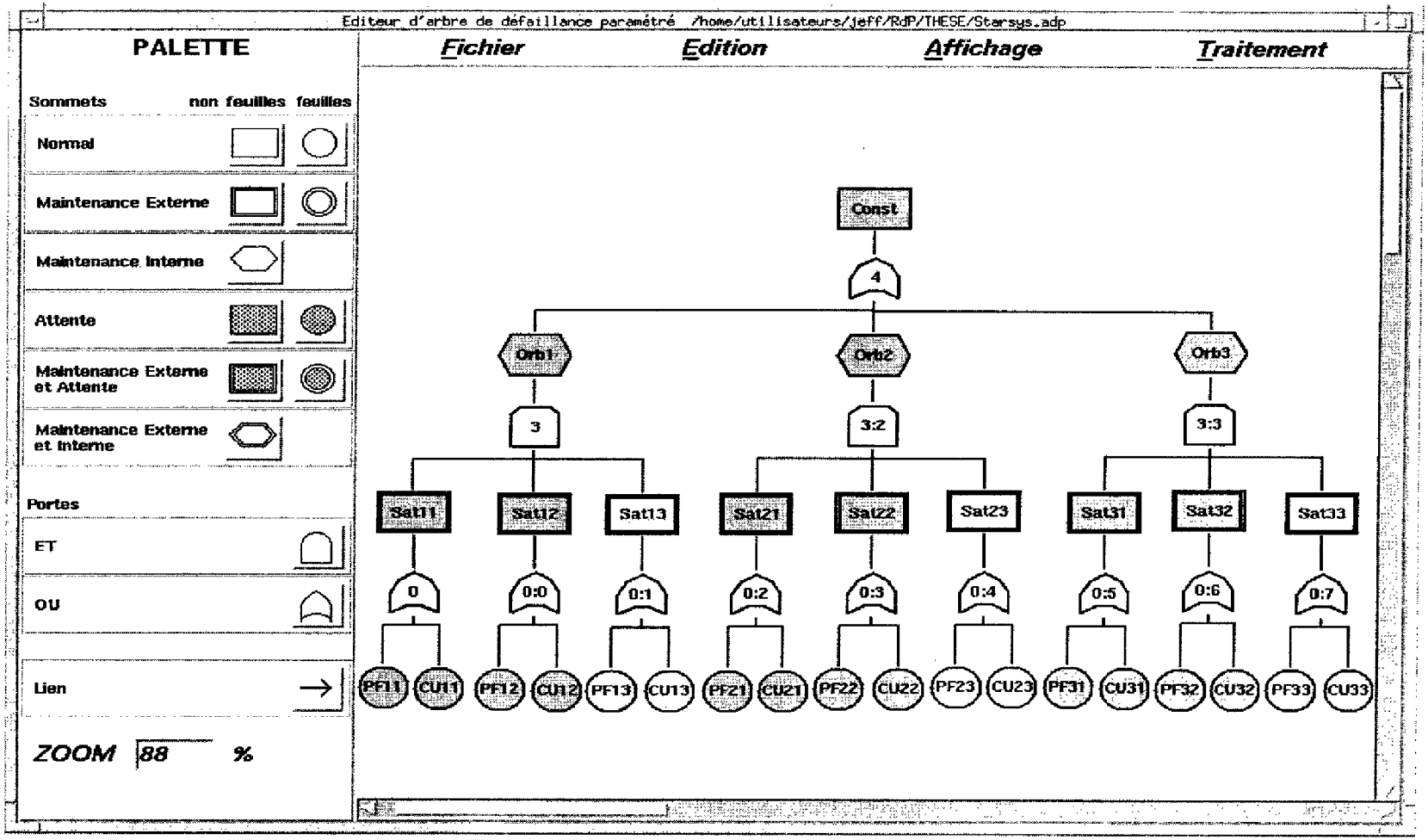


Figure VII.9 : Arbre de Défaillances Paramétrés décrivant le Segment Spatial de la Constellation

La redondance *2* parmi *3* entre les satellites de chaque plan d'orbite est modélisée :

- par le type de sommet (losange) associé à chaque orbite,
- par la mode attente affecté à l'un des satellites de chaque plan ainsi qu'à descendance (couleur différente)

La maintenance externe des satellites est décrite par le double rectangle associé à leur sommet.

VII.4.3. Edition des Paramètres de chaque Sommet de l'ADP

La Figure VII.10 montre l'édition des différents types de sommet de l'arborescence. On constate que l'éditeur de sommet ne propose que les paramètres licites pour le type du sommet dont on demande l'édition (obtenue en double-cliquant sur ce sommet). Il est par ailleurs possible de changer ce type en cliquant sur le bouton *Changer*.

Le premier éditeur présente la racine. Aucun paramètre n'est renseigné, donc aucun événement externe n'est à considérer à ce niveau.

Le second éditeur présente un sommet *plan d'orbite* de type *Maintenance Interne*. L'événement externe *Commutation* doit être pris en compte c'est pourquoi le paramètre correspondant est automatiquement proposé et doit être renseigné. Cette commutation, lorsqu'elle sera requise, interviendra après une durée fixe de *3* semaines (*dirac(504h)*).

Le troisième éditeur présente un sommet *satellite*. Comme il est de type *Maintenance Externe*, le paramètre *réparation* est proposé. On a choisi ici une réparation modélisée par un *réseau ressource* dont l'accès au fichier est décrit. De plus, on prend en compte l'initialisation du système à ce niveau (donc l'accès au fichier *réseau ressource* est renseigné).

Le dernier éditeur présente un sommet *charge utile*. C'est un sommet feuille pour lequel un des ascendant est de type *Maintenance Interne*, les paramètres *défaillance* et *défaillance en attente* sont donc proposés. On a choisi de décrire l'événement externe associé par une loi exponentielle.

VII.4.4. Génération Automatique de l'Arbre de Défaillances Dynamique

La génération automatique d'*Arbres de Défaillances Dynamiques* est obtenue à partir du menu *Traitement/Générer* de l'éditeur. *MISS-RdP* étant l'éditeur et le simulateur de réseaux de Petri utilisé au CNES et à ALCATEL ESPACE, les réseaux des *ADD* ainsi construits sont décrits au format de fichier de cet outil. Mais il reste tout à fait envisageable de pouvoir générer les *ADD* à d'autres formats de fichier comme par exemple celui de *SURF2*.

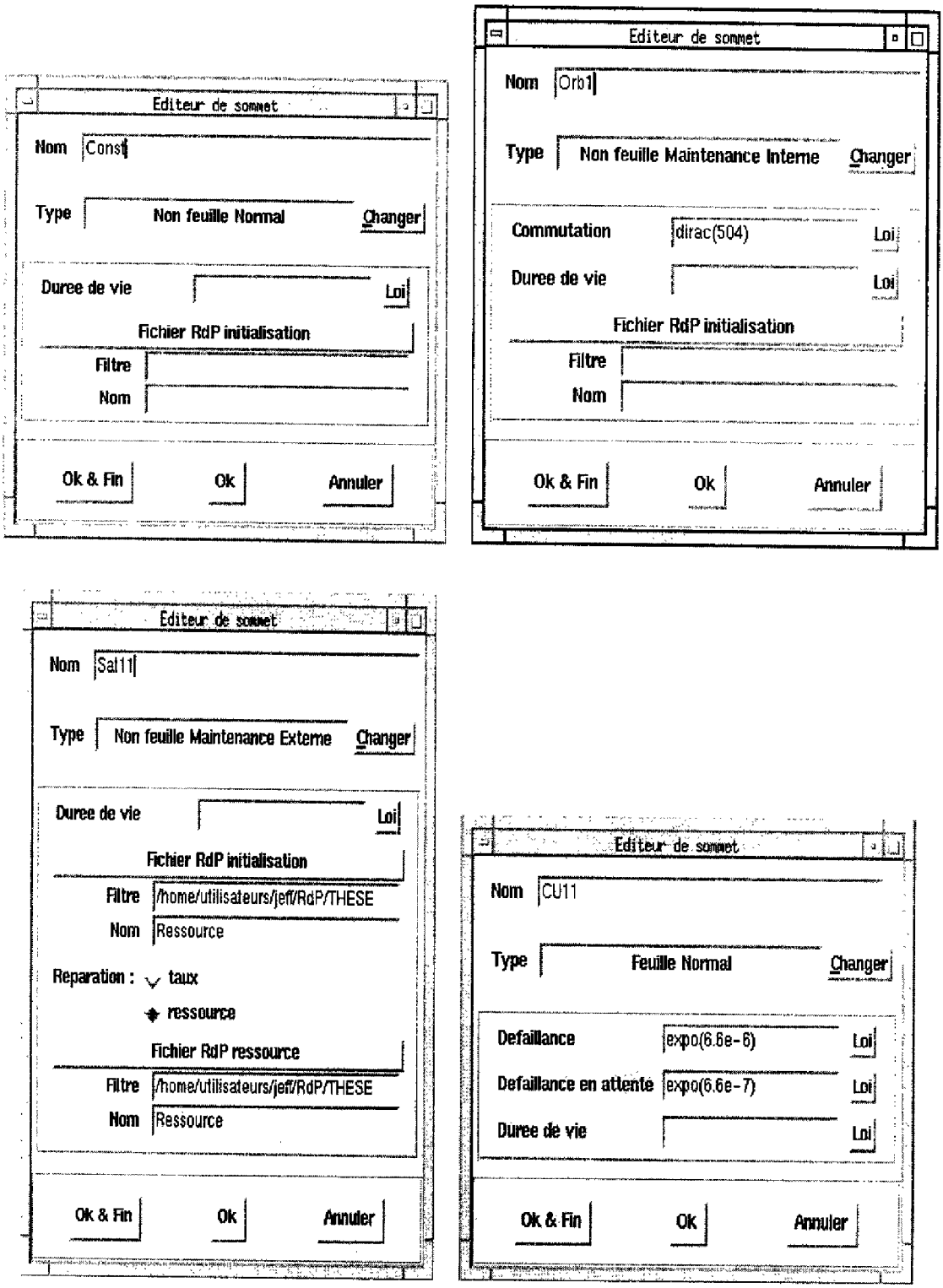


Figure VII.10 : Paramétrage Associé à chaque Sommet

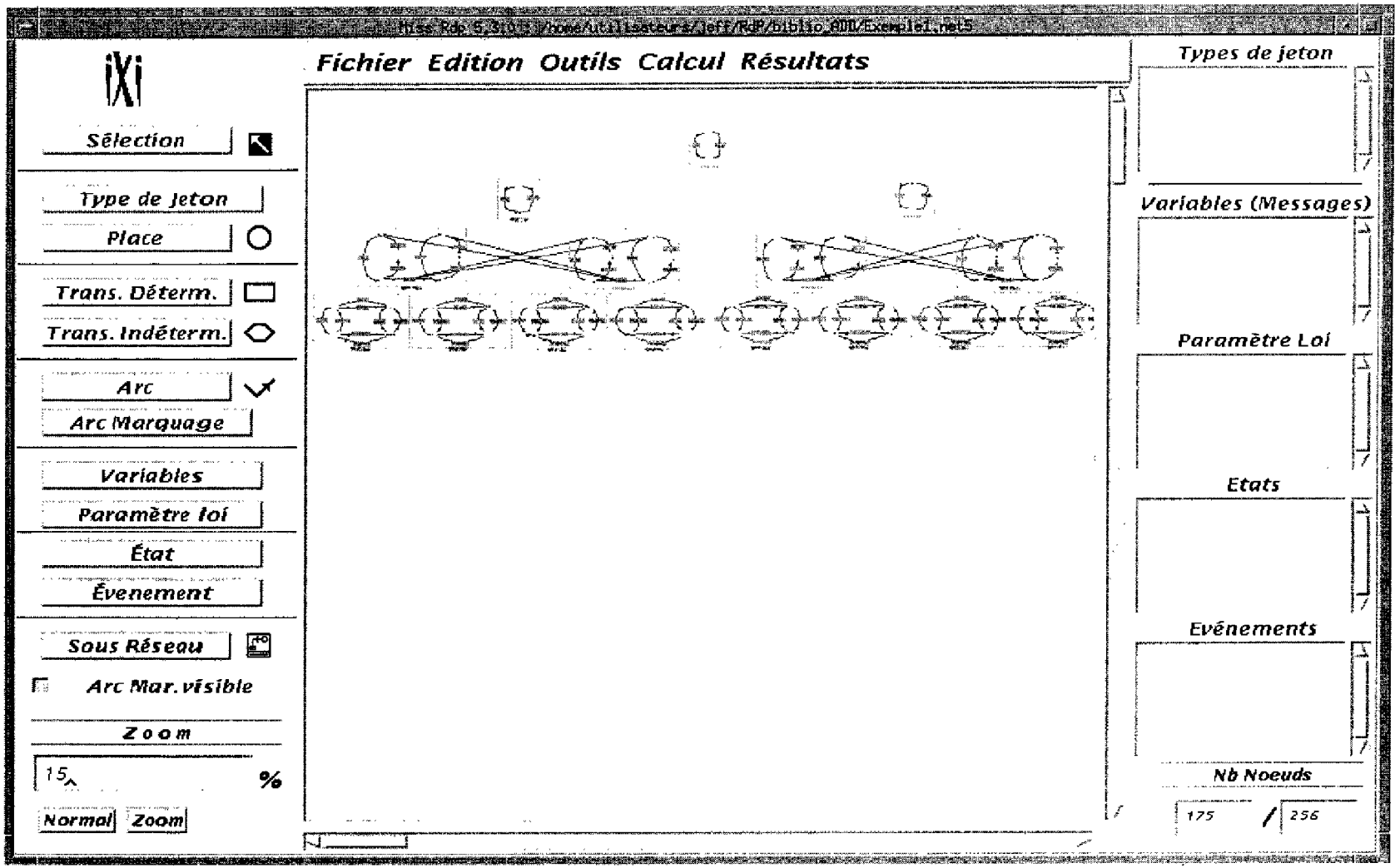


Figure VII.11 : *Arbre de Défaillances Dynamique* Généré pour l'*ADP* de la Figure VII.9

La Figure VII.11 montre une capture d'écran de *MISS-RdP* . L'*ADD* édité est celui généré à partir de l'*ADP* de la constellation décrit ci-avant mais pour lequel on a fait les simplifications suivantes :

- On ne considère que *2* plans d'orbite,
- La redondance passive entre les satellites d'un même plan est simple,
- Il n'y a pas d'initialisation,
- la maintenance externe des satellites est simplement décrite par un taux de réparation.

On reconnaît les *4* niveaux de l'arborescence. Pour le niveau *3* on voit des liens entre réseaux frères qui correspondent à la fusion des transitions *Commutation* c'est-à-dire à la synchronisation horizontale de ces réseaux.

VII.5. Conclusions

Ce dernier chapitre nous a permis de proposer la génération *d'Arbres de Défaillances Dynamiques* à partir *d'Arbres de Défaillances Paramétrés* comme principe d'aide à la modélisation de *modèles structurels* de Disponibilité Opérationnelle.

Cette approche présente l'avantage de réduire l'effort de modélisation à la construction d'*Arbres de Défaillances Paramétrés*. Ces derniers sont très simples d'emploi et déjà partiellement connus de nombreux fiabilistes.

Les *Arbres de Défaillances Dynamiques* générés sont des arborescences de réseaux de Petri dont les principes ont été explicitement définis au chapitre précédent : principes structurels grâces aux synchronisations internes verticales et horizontales et principe d'évolution grâce à l'algorithme de micro-évolution structurée.

L'aide à la construction de modèles de Sûreté de Fonctionnement basés sur les réseaux de Petri est un thème qui a suscité un intérêt important ces dernières années. C'est ainsi que, parallèlement à notre travail, nous avons eu connaissance d'autres approches développées.

La plus voisine de celle que nous avons proposée est certainement celle d'Olivier Daniel [Daniel 95] dédiée à l'étude des attributs de SdF des systèmes manufacturiers. Elle est basée sur la génération de Réseaux de Petri Stochastique à Synchronisations Internes. Ce modèle correspond à la fusion des possibilités de calcul Markovien des réseaux Stochastiques Généralisés [Ajmone 87] et des possibilités de modélisation des réseaux de Petri Synchronisés. Dédiée aux systèmes de production, cette approche propose, à l'instar des *Défaillances Dynamiques*, la génération automatique de modèle à partir de la traduction d'un formalisme très simple décrivant le problème (graphes de flux) et la composition de modèles génériques et prédéfinis. Elle tire partie de cette composition de modèles pour augmenter l'efficacité du calcul analytique des attributs de Sûreté de Fonctionnement. Différence majeure avec notre approche, elle est orientée calcul Markovien et n'envisage pas la décomposition *modèles structurels* et *modèles ressources*.

Plus anciennement Roland Lepold de SIEMENS [Lepold 92] a proposé un mécanisme de construction automatique de *RdPSG* pour l'étude de la performabilité des systèmes informatiques.

Plus récemment enfin, Manish Malhotra [Malhotra 95] proposait des algorithmes de conversion d'Arbres de Défaillances en réseaux de Petri Stochastiques et en réseaux Stochastiques de récompense (Stochastic Reward Nets). L'objectif étant, comme dans le cadre ici proposé, de bénéficier de la simplicité des modèles combinatoires tout en permettant de prendre en compte les phénomènes qui dépassent leur pouvoir de modélisation.

Conclusion

Les trois parties de ce mémoire ont exposé les directions suivies au cours de notre travail dans le but de favoriser l'utilisation des réseaux de Petri pour l'étude de la Disponibilité Opérationnelle des avant-projets spatiaux.

La première partie, après avoir décrit le contexte particulier de conception des systèmes spatiaux et les objectifs de ce type d'études, nous a permis de mettre en évidence les limites des approches classiques (modèles Combinatoires ou Etats-Transitions) face aux complexités que présentent les systèmes actuels.

La présentation détaillée des concepts théoriques fondamentaux des réseaux de Petri, sur la base d'exemples simples et familiers des fiabilistes et tant pour la modélisation que pour le traitement quantitatif des modèles, nous a donné la possibilité de souligner tous les attraits que proposé cette théorie face aux limitations évoquées. Ceci a par la suite était illustré sur un cas d'étude pratique de Disponibilité Opérationnelle du segment spatial d'une constellation de satellites.

Enfin, la dernière partie a présenté les bases d'une approche d'aide à la construction de *modèles structurels* décrivant les interactions entre les éléments d'un système du point de vue des fins de vie, des défaillances et des maintenances. La génération automatique d'*Arbre de Défaillances Dynamiques* permet de construire à partir d'un formalisme très simple des modèles basés sur les réseaux de Petri.

Contribution

La principale contribution de ce travail est d'avoir favoriser l'utilisation des réseaux de Petri dans un contexte applicatif nouveau. La participation active aux études de Disponibilité Opérationnelle menées sur des projets concrets nous a permis de convaincre les ingénieurs système et de guider les fiabilistes dans l'usage des réseaux de Petri. Forts d'une connaissance précise du domaine applicatif, nous avons pu rechercher et isoler les caractéristiques des réseaux de Petri essentielles à ce type d'études. Afin d'assister les fiabilistes dans la tâche de modélisation, nous avons proposé les bases d'une approche d'aide à la construction de modèles et illustré cette dernière grâce à un prototype logiciel.

Perspectives

Pour devenir opérationnelle, notre approche a besoin d'être affinée à plusieurs niveaux. Trois domaines sont essentiels :

1. L'aide à la construction de *modèle ressources*
2. Le couplage de *modèles ressources* aux *Arbres de Défaillances Dynamiques*
3. L'exploitation quantitative du modèle global obtenu.

Dans ce qui suit nous donnons quelques directions envisageables.

1. Aide à la construction des Modèles Ressources

Les *modèles ressources* ont pour but de décrire finement une action décrite sous forme de transition externe dans les *Arbres de Défaillances Dynamiques*. Par exemple, une transition *maintenance externe* pourrait être un processus complexe mettant en oeuvre parallélisme synchronisation et partage de ressources. Les techniques d'affinement de transition [Valette 76] et/ou de places [Murata 89] semblent donc une direction privilégiée dans l'aide à la construction de ces modèles.

2. Couplage avec des Modèles Ressources / Arbres de Défaillances Dynamiques

Le formalisme réseaux de Petri commun aux *Arbres de Défaillances Dynamiques* et aux *modèles ressources* permet d'envisager de façon simple le afin d'obtenir le modèle global d'un système. La Figure VIII.1 présente un couplage possible avec un *réseau ressource* décrivant une procédure de maintenance externe: il s'agit d'une communication par fusion de places entre les deux modèles (ou communication par sémaphores). La transition *Maint. Ext* du réseau *R(X)* est remplacée par une séquence de transitions internes non synchronisées. Le tir de la première transition génère un jeton dans la place intermédiaire *Att* et un jeton dans la place *In* partagée avec le *réseau ressource*. Ce dernier peut le prélever et effectuer le traitement correspondant à la procédure de maintenance externe. Lorsqu'il a terminé, il dépose un jeton dans la place *Out* qu'il partage avec *R(X)*. *R(X)* est ainsi informé que la maintenance externe est achevée.

Un couplage multiple peut également être envisagé basé sur ce même principe de fusion des places *In* et *Out* comme l'illustre la Figure VIII.2. Par exemple, les réseaux décrivant les états possibles d'un satellite pourraient partager le même réseau ressource pour décrire le processus de remplacement du satellite (comme c'est le cas au Chapitre V).

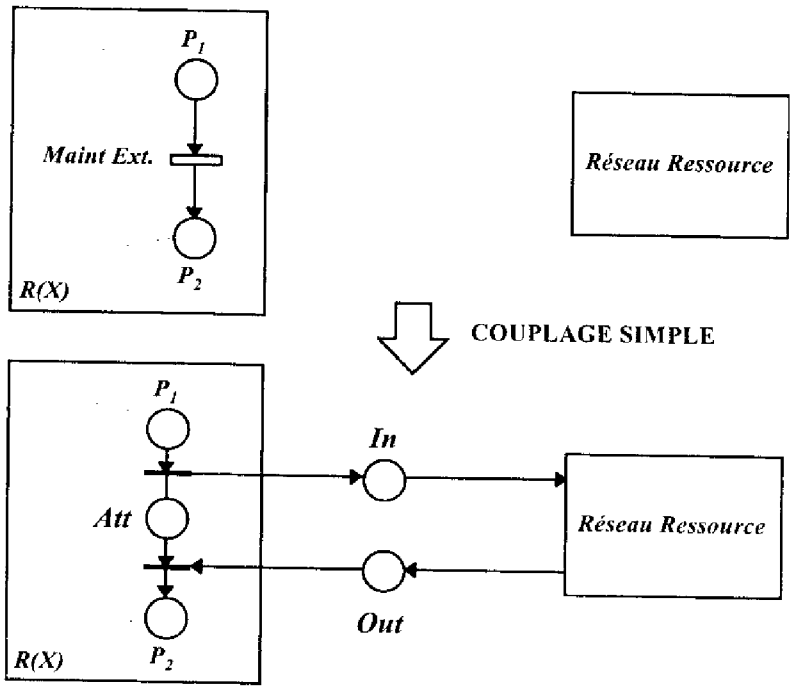


Figure VIII.1 : Principe du Couplage Simple entre un Réseau d'un *ADD* et un *réseau Ressource*

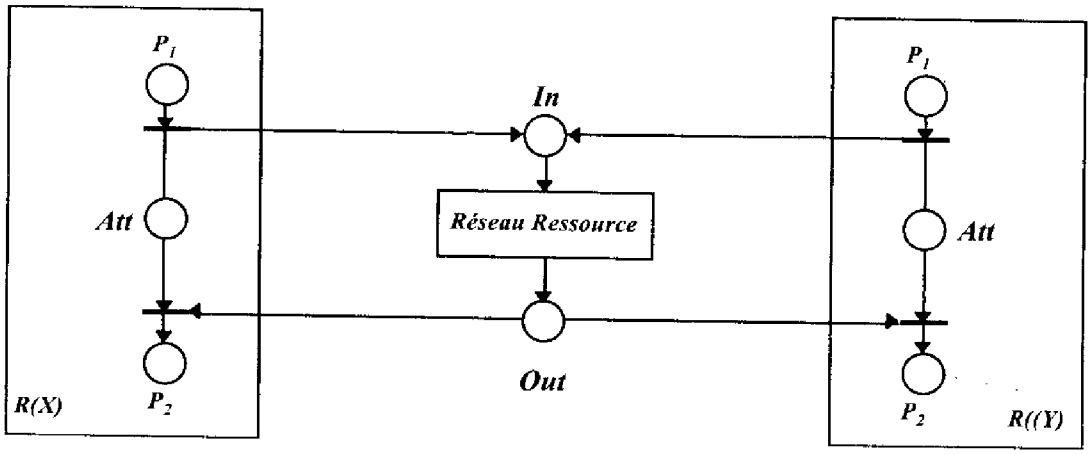


Figure VIII.2 : Principe du Couplage Double (ou Multiple)

3. Exploitation Quantitative

L'exploitation quantitative des *ADD* doit être possible, dans les cas simples, par calcul Markovien. Pour cela il faudrait développer un algorithme permettant de générer le graphe des états stables atteignables. C'est en effet à partir de ce dernier que l'on pourrait générer la matrice Etats-Transitions du modèle donnant accès au calcul Markovien.

Pour une exploitation du modèle global par simulation deux algorithmes d'évolution sont à considérer :

- L'algorithme de micro-évolution structurée permettant de faire évoluer l'arborescence des réseaux d'un ADD.
- L'algorithme d'évolution des réseaux de Petri Stochastiques pour l'évolution des réseaux ressources.

Bien entendu ces deux algorithmes devraient communiquer par échanges de messages ce qui justifie les couplages entre modèles envisagés.

Bibliographie

[Agard 68] J. Agard, *Les Méthodes de Simulation*, Monographies de Recherche Opérationnelle n°7 Dunod, 1968.

[Ajmone 84] M. Ajmone-Marsan, G. Conte, G. Balbo, *A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems*, ACM Transactions on Computer Systems 2, no 2: 93-122, 1984.

[Ajmone 87] M. Ajmone Marsan, G. Balbo, G. Chiola, and G. Conte,*Generalized Stochastic Petri Nets Revisited : Random Switches and Priority*, IEE, 1987.

[Atamna 94] Y. Atamna, *Réseaux de Petri Temporisés Stochastiques Classiques et Bien Formés: Définition, Analyse et Application aux Systèmes Distribués Temps Réel*, Thèse de Doctorat de l'Université Paul Sabatier, Toulouse, Octobre 1994.

[Bako 90] B. Bako, *Mise en Oeuvre et Simulation du Niveau Coordination de la Commande des Ateliers Flexibles, Une Approche Mixte Réseaux de Petri et Système de Règles*, Thèse de Doctorat de l'Université Paul Sabatier Toulouse, Octobre 1990.

[Berthelot 86] G. Berthelot, *Transformations and Decomposition of Nets*, LNCS 254, Springer-Verlag, 1986.

[Boyd 96] M. A. Boyd, *What Markov Modeling Can Do for You: An Introduction*, Tutorial Notes of Annual Reliability and Maintainability Symposium, Las Vegas, USA, January 1996.

[Brams 83] BRAMS, G. W., *Réseaux de Petri: Théorie et Pratique*, Edition Masson, Paris, 1983.

[Carter 92] D. E. Carter & B. S. Baker, *Concurrent Engineering*, Addison-Wesley, 1992.

[CEI 91] Commission Electrotechnique Internationale, *Gestion de la Sûreté de Fonctionnement*, Norme Internationale CEI 300-3-1, Première édition, 1991.

[Chiola 91] G. Chiola, *GreatSPN 1.5 Software Architecture*, 5th In. Conf. on Modeling Techniques and Tools for Computer Performance Evaluation, Torino, Italy, February 1991.

[Ciardo 93] G. Ciardo, *Generalized and Deterministic Stochastic Petri Nets*, Tutorials Notes of the Fifth PNPM, Toulouse, October 1993.

[Courtois 77] P. J. Courtois, *Decomposability, Queueing and Computer Science Applications*, Academic Press, New York, 1977.

[Daniel 95] O. Daniel, *Les Réseaux de Petri Stochastiques pour l'Evaluation des Attributs de la Sûreté de Fonctionnement des Systèmes Manufacturiers*, Thèse de l'Institut National Polytechnique, Grenoble, Janvier 1995.

[David 92] David R. et H. Alla, *Du Grafcet au Réseaux de Petri*, 2ème édition revue et corrigée, Hermès, Paris, 1992.

[Dosière 95] F. Dosière, *Disponibilité des Services d'un Réseau de Télécommunications par Satellites en Orbites Basses*, Thèse de l'Ecole Nationale Supèrieure des Télécommunications, Toulouse, Novembre 1995.

[Doyon 89] G. Doyon, *Systèmes et Réseaux de Télécommunication en Régime Stochastique*, Collection Technique et Scientifique, des Télécommunications, Edition Masson, 1989.

[Dugan 92] J. B. Dugan, S. J. Bavuso and M. A. Boyd, *Dynamic Fault-Tree Models for Fault-Tolerant Computer Systems*, IEE Transactions on Reliability, VOL. 41, N°3, September 1992.

[Dugan 96] J. B. Dugan and S. A. Doyle New *Results in Fault-Tree Analysis*, Tutorial Notes of Annual Reliability and Maintainability Symposium, Las Vegas, USA, January 1996.

[Ereau 93] J. F. Ereau, *Réseaux de Petri pour la Modélisation d'une Constellation de Satellites*, Rapport de D.E.A. Automatique Informatique Industrielle, Université Paul Sabatier, Toulouse, Juin 1993.

[Ereau 93 (2)] J. F. Ereau, *Etudes de Disponibilité des Systèmes TPFO et BIMILSAT*, Rapport de prestation de Services LAAS pour le CNES, Toulouse, Août 1993.

[Ereau 94] J. F. Ereau, M. Saleman, R. Valette et Hamid Demmou, *Réseaux de Petri pour l'Evaluation des Systèmes Redondants*, Colloque ESREL' 94 et λμ9, La Baule, Juin 1994.

[Ereau 94 (2)] J.F. Ereau et R. Blondel, *Disponibilité du Segment Spatial du Système GLOBALSTAR*, Rapport Interne ALCATEL ESPACE, Toulouse, 1994.

[Ereau 95] J. F. Ereau, H. Demmou et M. Saleman, *Dynamic Fault Trees Based on Synchronized Petri Nets*, 7th European Simulation Symposium, Erlangen, Germany, October 1995.

[Ereau 95 (2)] J. F. Ereau, *Arbres de Défaillances Dynamiques Basés sur les Réseaux de Petri Synchronisés*, Rapport interne CNES, Juin 1995.

[Ereau 95 (3)] J.F. Ereau et M. Saleman, *Disponibilité du Segment Spatial du Système STARSYS*, Rapport Interne CNES N° CT/AQ/SE/SF 95-856, Toulouse, Septembre 1995.

[Ereau 96] J. F. Ereau et M. Saleman, *Modeling & Simulation of a Satellite Constellation based on Petri Nets*, Annual Reliability and Maintainability Symposium, Las Vegas, USA, January 22-26, 1996.

[Ereau 97] J. F. Ereau, M. Saleman, R. Valette et Hamid Demmou, *Petri Nets for the Evaluation of Redundant Systems*, Reliability Engineering and System Safety 55, Elsevier Science Limited, 1997.

[ESA 95] European Space Agency, ECSS-Q-30-Draft 5, 23 Octobre 1995.

[Florin 90] G. Florin, C. Fraize et S. Natkin, *Stochastic Petri Nets: Properties Applications and Tools*, Rapport de Recherche, CEDRIC-CNAM N°90-05, 1990.

[Gory 92] J. F. Gory, J. Chevalier et J. Michaud, *S2000+ ou la Conception des Systèmes Spatiaux en l'An 2000*, Symposium sur la Conception en l'an 2000 et Au-Delà - Outils et Technologies, Strasbourg, Novembre 1992.

[IXI 95] IXI, *MISS-RdP version 5, Manuel de l'utilisateur*, Toulouse, 1995.

[Juanole 91] G. Juanole and Y. Atamna, *Dealing with Arbitrary Time Distributions with the Stochastic and Timed Petri Net Model -Application to Queueing Systems*, Proceedings of PNPM, 1991.

[Kemeny 76] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*, Springer-Verlag, 1976.

[Lahorgues 95] C. Lahorgues et J. F. Ereau, *ADP : Un Editeur d'Arbres de Défaillances Paramétrés*, Rapport Interne CNES, Toulouse, Octobre 1995.

[Laprie 85] J. C. Laprie, *Dependable Computing and Fault Tolerance : Concepts and Terminology*, Proceedings of 15th International Symposium, Fault Tolerant Computing, 1985.

[Larson 92] W. J. Larson et J. R. Wertz, *Space Mission Analysis and Design Process*, in Space Mission Analysis and Design, Space Technology Library, Microcosm Inc., 1992.

[Law 91] A. Law, W. D. Kelton, *Simulation Modeling & Analysis*, Mc Graw-Hill Inc, New-York, USA, 1991.

[Lepold 92] R. Lepold, *Performability Evaluation of Degradable Computer Systems Based on Stochastic Petri Nets*, Thèse de Doctorat de l'Université de Haute Alsace Mulhouse, France April 1992.

[Leroudier 80] J. Leroudier, *La Simulation à Evénements Discrets*, Monographies d'Informatique de l'Afcet, Editions Hommes et Techniques, 1980.

[Malhotra 95] M. Malhotra and K. Trivedi, *Dependability Modeling Using Petri Nets*, IEE Transactions on Reliability, VOL. 44, N°3, September 1995.

[Malhotra 94] Malhotra M. and K. Trivedi, *Power-Hierarchy of Dependability-Model Types*, IEEE Transactions on Reliability 43, no. 3: 493-502, September 1994.

[Moalla 78] M. Moalla, J. Pulou et J. Sifakis, *Réseaux de Petri Synchronisés*, R.A.I.R.O. Automatique / Systems Analysis and Control, Vol. 12, N°2 1978.

[Molloy 81] M. K Molloy, *On the Integration of Delay and Throughput Measures in Processing Models*, Ph. D. Thesis, University of California, Los Angeles, USA, 1981.

[Molloy 82] M. K. Molloy, *Performance Analysis using Stochastic Petri Nets*, IEE Transactions on Comuters, September 1982.

[Moyse 95] F. Moyse, *Précision des Résultats de Simulation du Logiciel MISS-RdP*, Rapport de stage de DEA de Mathématiques Appliquées, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace & Université Paul Sabatier, Toulouse, 1995.

[Murata 89] T. Murata, *Petri Nets: Properties, Analysis and Applications*, Proceedings of the IEEE, Vol. 77, N°4, April 1989.

[Natkin 80] S. Natkin, *Les Réseaux de Petri Stochastiques*, Thèse de Docteur Ingénieur CNAM, Paris, Juin 1980.

[Ousterhout 95] J. K. Ousterhout, *TCL & TK Toolkit*, User Manual, 1995.

[Pagès 80] A. Pagès et M. Gondran, *Fiabilité des Systèmes*, Eyrolles, 1980.

[Peterson 81] J. L. Peterson, *Petri Net and the Modelling of Systems*, Prentice Hall, 1981.

[Petri 66] C. A. Petri, *Communication with Automata*, Final Report, Volume 1, RADC TR-65-377 Applied Data Research, Princeton, NJ, USA, 1966.

[Raspaud 96] L. Raspaud et J.F. Ereau, *Génération Automatique d'Arbres de Défaillance Dynamiques*, Rapport Interne CNES, Toulouse, Mai 1996.

[Riebman 88] A. L. Reibman and K. S. Trivedi, *Numerical Transient Analysis of Markov Models*, Computers and Operations Research, VOL 15, N°1, pp. 19-36, 1988.

[Rouziès 94] C. Rouziès, *La Sûreté de Fonctionnement*, Techniques et Technologies des Véhicules Spatiaux, CNES - Cépaduès Editions, 1994.

[Saleman 94] M. Saleman, Techniques et Technologies des Véhicules Spatiaux, CNES - Cépaduès Editions, 1994.

[Saleman 96] M. Saleman and J.F. Ereau, *Petri Nets for a Space Operational Availability Study*, Advanced and Technology Workshop, July 1996, Toulouse.

[Sibertin 85] C. Sibertin Blanc, *Petri Nets with Data Structure*, 6th European Workshop on Petri Nets and Applications, June 1985, Espoo, Finland.

[Schmitter 93] E. Schmitter, *The Implications of Industrial Systems Complexity on Methodologies for System Design and Evaluation*, Invited Talk, 5th International Workshop on Petri Nets and Performance Models, Toulouse, October 1993.

[Shooman 70] Shooman M., *The Equivalence of Reliability Diagram and Fault-Tree Analysis*, IEEE Transactions on Reliability 19, pp 469-473, May 1970.

[Subias 94] M. Subias, *Développement d'un Véhicule Spatial*, Techniques et Technologies des Véhicules Spatiaux, CNES - Cépaduès Editions, 1994.

[SURF-2 93] LAAS-CNRS et CEP-Systèmes, *SURF-2 Manuel de l'Utilisateur*, 1993.

[TOMSPIN 92] G. Klas, R. Lepold, *TOMSPIN, a Tool for Modeling with Stochastic Petri Nets*, pp 618-623, CompEuro 92, La Hague, May 1992.

[UltraSAN 95] Center for Reliable and High Performance Computing, *UltraSAN User's Manual Version 3.0*, Coordinated Science Laboratory, University of Urbana-Champaign, 1995.

[Valette 76] R.Valette, *Sur la Description, l'Analyse et la Validation des Systèmes de Commande Parallèles*, Thèse d'État de l'Université Paul Sabatier, Toulouse, 1976.

[Valette 79] R. Valette, *Analysis of Petri Nets by Stepwise Refinements*, J. Compt. Sci., VOL 18, N°1, February 1979.

[Valette 92] R. Valette, *Les Réseaux de Petri*, Support de cours, Toulouse, France, Mai 1992.

[Villemeur 88] A. Villemeur, *Sûreté de Fonctionnement des Systèmes Industriels*, Eyrolles, 1988.

Thèse de Jean-François EREAU

« Réseaux de Petri pour l'Etude de la Disponibilité Opérationnelle des Systèmes Spatiaux en Phases d'Avant-Projet »

RESUME

Si certains conservent un objectif scientifique et expérimental, la plupart des projets spatiaux visent maintenant à développer des systèmes profitables soumis à de fortes contraintes opérationnelles. Dimensionner ces systèmes, « ni trop, ni trop peu », est donc un objectif majeur qui conditionne leur viabilité économique. Si trop de risques sont pris les investisseurs se feront rares, et si le système est surdimensionné son coût peut être également rapidement dissuasif.
Dans ce contexte, l'étude de la Disponibilité Opérationnelle en phases d'avant-projet donne certains critères précieux pour évaluer le compromis coût / prise de risque et permet ainsi de guider très tôt certains choix de conception. Cependant la complexité et la taille croissante de ces systèmes ont rapidement mis en évidence certaines limites des méthodes classiques d'évaluation de cette grandeur.
La théorie des réseaux de Petri, riche d'une trentaine d'années de recherche et principalement appliquée à l'analyse, l'évaluation et la commande des systèmes distribués, offre des perspectives intéressantes pour dépasser ces limites.
La principale contribution de ce travail mené au sein de l'Agence française de l'Espace et d'un industriel du spatial a été de favoriser son utilisation pour l'étude de la Disponibilité Opérationnelle de système spatiaux complexes et dans le cadre très concret des phases d'avant-projets. On a tout d'abord justifié l'intérêt que les réseaux de Petri présentent tant pour la modélisation que pour l'évaluation de tels systèmes par comparaison aux approches classiques. Puis on a illustré leur utilisation sur des programmes bien réels et dans le cadre d'un travail intégré avec les équipes projets. Enfin, on a proposé les bases d'une démarche de modélisation orientée application dont le but est d'aider des non spécialistes à concevoir aisément des modèles de systèmes complexes. Ce sont ces trois grandes étapes qui sont ici présentées.

Mots Clefs : Réseaux de Petri - Disponibilité Opérationnelle - Systèmes Spatiaux - Aide à la Modélisation.

ABSTRACT

Most of nowadays space projects follow economic goals and are submitted to strong operational constraints. Avoiding under (or over) sizing up of these systems is a main objective which can decide if the project can be realized or not. If too much risks are taken, then investment will be very difficult to find, and, if the system is over sized up the price will dramatically increase and be dissuasive.
In that context, the study of Operational Availability in first design phases can provide precious criteria in order to estimate the compromise cost / risk and then can help to make early important design choices. However complexity and size of these systems increase in such a way that traditional availability approaches can no more be useful.
Petri net theory, rich of thirty years worldwide research and principally devoted to the analysis, evaluation and control of distributed systems, offers stimulating perspectives to overcome these limits.
The main contribution of this work, led inside French space agency and industry, was to promote its use for Operational Availability study of complex space systems in the concrete context of early design phases. We have firstly justified Petri nets interest for modeling and evaluating such systems by comparison with traditional approaches. Then, we have illustrated their use over real space program studies in the context of integrated work with project teams. At least, we have proposed the basis for a modeling approach, application oriented, which aim is to help non Petri net addicts to build easily models for complex systems. These main three steps are presented here.

Key Words : Petri Nets - Operational Availability - Space Systems - Modeling Approach